

# CROSSTALK

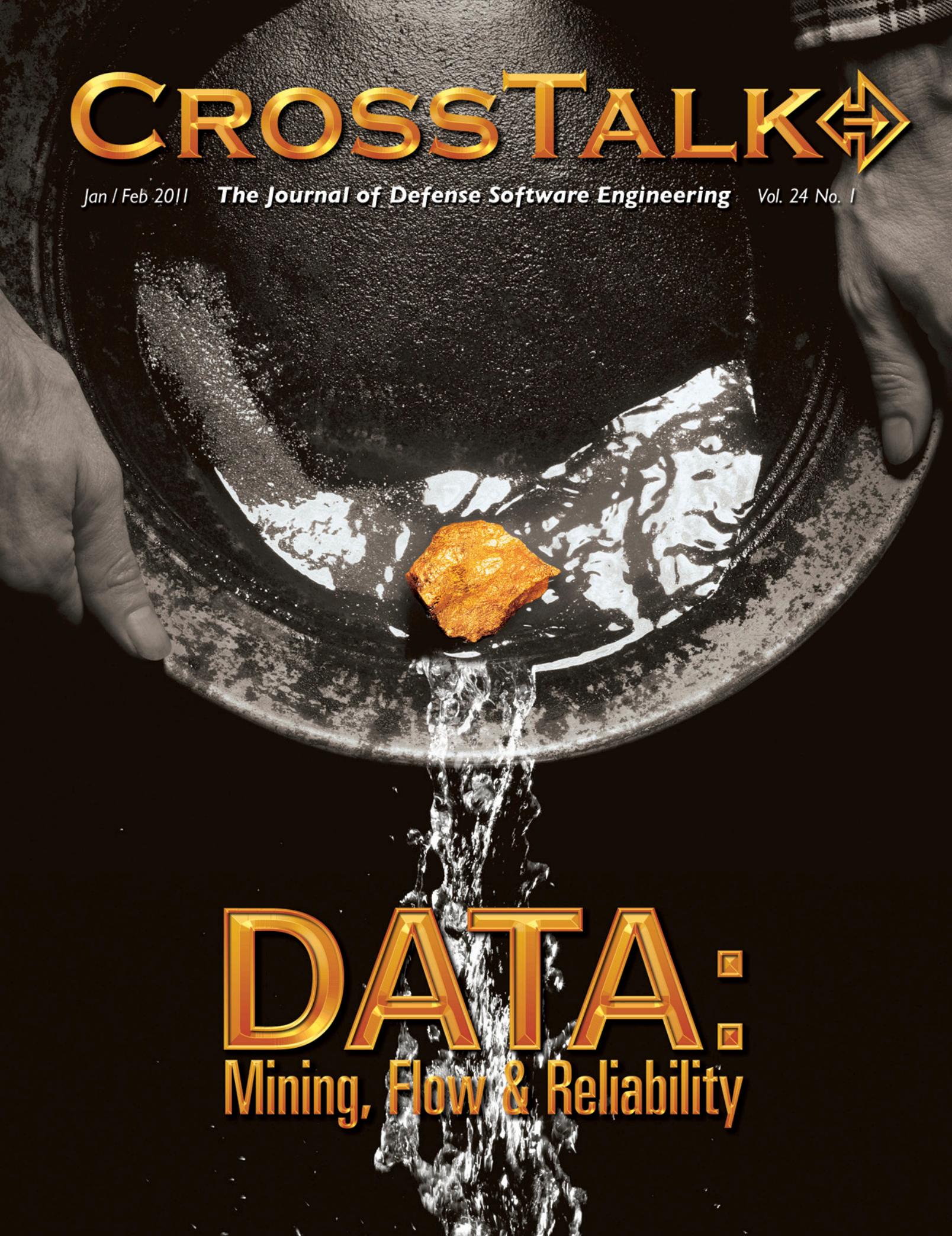
Jan / Feb 2011

*The Journal of Defense Software Engineering*

Vol. 24 No. 1

# DATA.

Mining, Flow & Reliability



## Report Documentation Page

*Form Approved  
OMB No. 0704-0188*

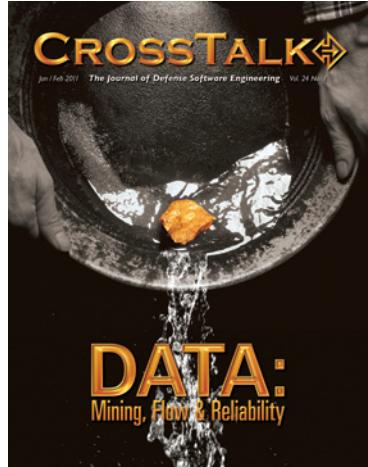
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE <b>JAN 2011</b>	2. REPORT TYPE	3. DATES COVERED <b>00-01-2011 to 00-02-2011</b>		
4. TITLE AND SUBTITLE <b>CrossTalk. The Journal of Defense Software Engineering. Volume 24, Number 1, Jan/Feb 2011</b>			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>517 SMXS MXDEA,6022 Fir Ave,Hill AFB,UT,84056-5820</b>			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>32</b>
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>		

## Departments

3 Sponsor's Note

28 BackTalk



Cover Design by Kent Bingham

## DATA MINING

## 6 Interview with Dr. Randall W. Jensen

Cost estimation guru, Dr. Randall Jensen, gives his insight into tightening up project parameters.

## 10 Data Mining for Process Improvement

Data mining techniques can be used to filter many variables to a vital few to build or improve predictive models. Specific examples are provided in four categories: classification, regression, clustering, and association.

by Paul Below

## 16 Demystifying Cloud Computing

Is transitioning to the cloud right for your organization? What are the challenges involved with a cloud computing migration? A look at the pros, cons, terminologies of, and alternatives to cloud computing.

by Qusay F. Hassan

## 22 A Comparison of Parametric Software Estimation Models Using Real Project Data

Researchers and practitioners of software metrics have developed models to help project managers and system engineers produce estimates of project effort, duration, and quality.

by George Stark

## CROSSTALK

OSD (AT&amp;L) Stephen P. Welby

NAVAIR Jeff Schwalb

DHS Joe Jarzombek

309 SMXG Karl Rogers

Acting Publisher Justin T. Hill

Advisor Kasey Thompson

Article Coordinator Lynne Wade

Managing Director Brent Baxter

Managing Editor Brandon Ellis

Associate Editor Colin Kelly

Art Director Kevin Kiernan

Phone 801-775-5555

E-mail [stsc.customerservice@hillaf.mil](mailto:stsc.customerservice@hillaf.mil)CrossTalk Online [www.crosstalkonline.org](http://www.crosstalkonline.org)

## CROSSTALK, The Journal of Defense Software Engineering

is co-sponsored by the Under Secretary of Defense for Acquisition, Technology and Logistics (USD(AT&L)); U.S. Navy (USN); U.S. Air Force (USAF); and the U.S. Department of Homeland Defense (DHS). USD(AT&L) co-sponsor: Director of Systems Engineering. USN co-sponsor: Naval Air Systems Command. USAF co-sponsor: Ogden-ALC 309 SMXG. DHS co-sponsor: National Cyber Security Division in the National Protection and Program Directorate.

**The USAF Software Technology Support Center (STSC)** is the publisher of **CROSSTALK** providing both editorial oversight and technical review of the journal. **CROSSTALK's** mission is to encourage the engineering development of software to improve the reliability, sustainability, and responsiveness of our warfighting capability.

**Subscriptions:** Visit <[www.crosstalkonline.org/subscribe](http://www.crosstalkonline.org/subscribe)> to receive an e-mail notification when each new issue is published online or to subscribe to an RSS notification feed.

**Article Submissions:** We welcome articles of interest to the defense software community. Articles must be approved by the **CROSSTALK** editorial board prior to publication. Please follow the Author Guidelines, available at <[www.crosstalkonline.org/submit-guidelines](http://www.crosstalkonline.org/submit-guidelines)>. **CROSSTALK** does not pay for submissions. Published articles remain the property of the authors and may be submitted to other publications. Security agency releases, clearances, and public affairs office approvals are the sole responsibility of the authors and their organizations.

**Reprints:** Permission to reprint or post articles must be requested from the author or the copyright holder and coordinated with **CROSSTALK**.

**Trademarks and Endorsements:** This Department of Defense (DoD) journal is an authorized publication for members of the DoD. Contents of **CROSSTALK** are not necessarily the official views of, or endorsed by, the U.S. government, the DoD, the co-sponsors, or the STSC. All product names referenced in this issue are trademarks of their companies.

**CROSSTALK Online Services:**

For questions or concerns about [crosstalkonline.org](http://www.crosstalkonline.org) web content or functionally contact the CrossTalk webmaster at 801-417-3000 or [webmaster@luminpublishing.com](mailto:webmaster@luminpublishing.com).

**Back Issues Available:** Please phone or e-mail us to see if back issues are available free of charge.

**CROSSTALK** is published six times a year by the U.S. Air Force STSC in concert with Lumin Publishing <<http://www.luminpublishing.com>>.

**CROSSTALK** would like to thank the 309 SMXG for sponsoring this issue.

# Not Just Another New Year!



The January/February 2011 issue of **CROSSTALK** is not just the issue by which we'll start off another new year; it is the ushering in of a new era of **CROSSTALK**. By now you've seen the new web site, the new format, and the improved search capabilities. Reader feedback has been positive to say the least and we thank you for taking time to respond with your opinions. **CROSSTALK** will continue forward with our efforts to reach new mobile applications in the near future. Please be sure to add your input as to which devices **CROSSTALK** should target by clicking on the survey question on the home page at [crosstalkonline.org](http://crosstalkonline.org) or click [here](#) to go directly to the survey.

In the past year it was **CROSSTALK's** pleasure to published interviews with legends such as the late Watts Humphreys and Grady Booch. We thank both men for their contributions, posthumously and otherwise. It was an honor to add to Mr. Humphrey's published works as he was a 13-time contributor to **CROSSTALK**; he will be missed.

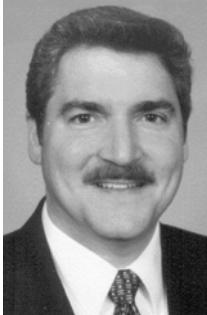
Our series of interviews continues with cost estimation guru, Dr. Randall Jensen. Dr. Jensen has spent the last 35 years gathering, filtering, and structuring data to use in cost estimation models. His lively comments will enlighten those looking to tighten up project parameters. Other articles this month discuss data in reference to cloud computing, estimating, and data mining for the purpose of process improvement. All-in-all, we hope this issue of **CROSSTALK** finds you well and that you have a prosperous and safe new year.

**Karl Rogers**  
Director 309 SMXG

# Announcing CROSSTALK's Co-Sponsor Team for 2011

I would like to once again express sincere thanks to the 2010 **CROSSTALK** co-sponsors. Simply put, **CROSSTALK** would not exist without them and their generous financial support. As Publisher, I receive countless kudos—via e-mail and the phone—expressing appreciation for an article or issue focus that contributed to individual or organizational success. These compliments really belong to the co-sponsors, who spark countless themes and help bring us the best authors in defense software engineering. Likewise, it is my pleasure to introduce **CROSSTALK's** 2011 co-sponsor team and offer profound gratitude for their continued support and commitment to this journal. I know firsthand of their vision, caring, and dedication to their industry and it is manifested through support of **CROSSTALK**. Each co-sponsor and their organization will assist our staff by lending us their inexhaustible experience in engineering, systems, security, acquisition, tools, processes, models, infrastructure, people, and (of course) software. Co-sponsor team members are identified in this section with a description of their organization. Please look for their contributions each month in our From the Sponsor column, found on page 3. Their organizations will also be highlighted on the back cover of each issue of **CROSSTALK**.

**Kasey Thompson**  
**CROSSTALK**



**Joe Jarzombek**  
*Department of Homeland Security (DHS) – Director of Software Assurance (SwA)*

The DHS National Cyber Security Division serves as a focal point for SwA, facilitating national public-private efforts to promulgate best practices and methodologies that promote integrity, security, and reliability in software development and acquisition. Collaborative efforts of the SwA community have produced several publicly available online resources. For more information, see the Build Security In Web site <<https://buildsecurityin.us-cert.gov>> and the SwA Community Resources and Information Clearinghouse <<https://buildsecurityin.us-cert.gov/swa>>. Both provide coverage of topics relevant to the broader stakeholder community.



which it produces tangible, external results for the Sailor and the Marine. First are its People that we develop and provide the tools, infrastructure, and processes needed to do their work effectively. Next is Current Readiness that delivers NAVAL aviation units ready for tasking with the right capability, at the right time, and the right cost. Finally is Future Capability in the delivery of new aircraft, weapons, and systems on time and within budget that meets Fleet needs and provides a technological edge over our adversaries. See <<http://www.navair.navy.mil>> for more information.

**Joan Johnson**  
*Naval Air Systems Command (NAVAIR), Systems Engineering Department – Director, Software Engineering*



**Stephen P. Welby**  
*Office of the Under Secretary of Defense for Acquisition, Technology and Logistics - Director, Systems Engineering*

The Systems Engineering directorate is the focal point for all policy, practice, and processes relating to DoD systems engineering and its key elements including technical risk management, software engineering, manufacturing and production, quality, standardization, system of systems engineering, and related disciplines. Offices in the directorate include Major Program Support, Mission Assurance, and Systems Analysis, with responsibilities in program support and oversight, systems engineering policy and guidance, human capital, systems integration, and program protection. The 2011 focus areas include workforce development, early systems engineering and pre-acquisition development planning, simplifying acquisition guidance, and promoting best systems engineering practices to reduce risk <<http://www.acq.osd.mil/se>>.



**Karl Rogers**  
*Director, 309 Software Maintenance Group (SMXG) at the Ogden-Air Logistics Center*

The 309 Software Maintenance Group (SMXG) at the Ogden-Air Logistics Center is a recognized world leader in cradle-to-grave systems support, encompassing hardware engineering, software engineering, systems engineering, data management, consulting, and much more. Their accreditations also include AS 9100 and ISO 9000. See <<http://www.309SMXG.hill.af.mil>> for more information.

## WANT TO BECOME A CO-SPONSOR?

**CROSSTALK** co-sponsors enjoy many benefits such as inclusion of a page-long co-sponsor's note, placement of their organization's logo on the back cover for 12 issues, placement of the Director's name and organization on each issues' masthead, special sponsorship references in various issues, the ability to provide authors from within their community in regard to their sponsored issue, and online placement on the **CROSSTALK** Web site.

**CROSSTALK** co-sponsors are also invited each year to provide direction for future **CROSSTALK** themes and feedback from the software defense community at large. Co-sponsors are also invited to participate in an annual meeting held during the Systems and Software Technology Conference to discuss emerging needs, trends, difficulties, and opportunities which **CROSSTALK** may address in an effort to best serve its readers.

**CROSSTALK** welcomes queries regarding potential sponsorship throughout the year. For more information about becoming a **CROSSTALK** co-sponsor, please contact Kasey Thompson at (801) 586-1037 or <[kasey.thompson@hill.af.mil](mailto:kasey.thompson@hill.af.mil)>.

# The 2011 CROSSTALK Editorial Board

**CROSSTALK** proudly presents the 2011 **CROSSTALK** Editorial Board. Each article submitted to **CROSSTALK** is reviewed by two technical reviewers from the following list. Their insights improve the readability and usefulness of the articles that we publish. We give a very special thanks to all those participating in our 2011 **CROSSTALK** Editorial Board.

Wayne Abba	Abba Consulting	Walter Krall	Independent
COL Ken Alford, Ph.D.	Brigham Young University	Theron Leishman	Northrop Grumman
Bruce Allgood	309th Software Maintenance Group	Glen L. Luke	309th Software Maintenance Group
Greg Anderson	Weber State University	Gabriel Mata	309th Software Maintenance Group
Brent Baxter	Software Technology Support Center	Jim McCurley	Software Engineering Institute
Jim Belford	OO-ALC Engineering Directorate	Paul McMahon	PEM Systems
Gene Bingue	U.S. Navy	Dr. Max H. Miller	Raytheon Integrated Defense Systems
Lt. Col. Christopher A. Bohn, Ph.D.	HQ Air Force Special Operations Command	Mark Nielson	Software Technology Support Center
Mark Cain	309th Software Maintenance Group	Mike Olsem	U.S. Army
Dr. Alistair Cockburn	Humans and Technology	Dr. Kenneth E. Nidiffer	Software Engineering Institute
Dr. David A. Cook	Stephen F. Austin State University	Doug J. Parsons	Army PEO Simulation, Training and Instrumentation
Rushby Craig	309th Software Maintenance Group	Tim Perkins	500 CBSS/GBLA (Atmospheric Early Warning System)
Greg Daich	SAIC	Gary A. Petersen	Arrowpoint Solutions, Inc.
Les Dupaix	Software Technology Support Center	Vern Phipps	Arrowpoint Solutions, Inc.
Paul Croll	Computer Science Corporation	David Putnam	309th Software Maintenance Group
Robert W. Ferguson	Software Engineering Institute	Brian Rague	Weber State University
Dr. Doretta Gordon	Southwest Research Institute	Kevin Richins	Arrowpoint Solutions, Inc.
Dr. John A. "Drew" Hamilton Jr.	Auburn University	Gordon Sleve	Robbins Gioia LLC
Gary Hebert	538th Aircraft Sustainment Group	Larry Smith	Software Technology Support Center
Tony Henderson	309th Software Maintenance Group	Dr. John Sohl	Weber State University
Lt. Col. Brian Hermann, Ph.D.	Defense Information Systems Agency	Elizabeth Starrett	OO-ALC Engineering Directorate
Lt. Col. Marcus W. Hervey	Air Force Institute of Technology	Tracy Stauder	309th Software Maintenance Group
Thayne Hill	309th Software Maintenance Group	COL John "Buck" Surdu, Ph.D.	Army Research, Development, and Engineering Control
George Jackelen, PMP	Software Consultants, Inc.	Dr. Will Tracz	Lockheed Martin Integrated Systems and Solutions
Dr. Randall Jensen	Software Technology Support Center	Jim Van Buren	Charles Stark Draper Laboratory
Alan C. Jost	Raytheon – Network Centric Systems	J. Bruce Walker	AFNWC
Daniel Keth	309th Software Maintenance Group	David R. Webb	309th Software Maintenance Group
Paul Kimmerly	U.S. Marine Corps	Drew Weidman	Weber State University
		Mark Woolsey	309th Software Maintenance Group
		David Zubrow	Software Engineering Institute

# Software Estimating, People, and the Smell of Popcorn:

An Interview with Dr. Randall Jensen



**CROSSTALK:** What are the foundational benefits of software cost estimating?

**Randy:** Without a software estimate—a cost and schedule estimate—you can't manage the program. Fundamentally, that's what the problem is. People estimate off the top of their heads very optimistically and, from the moment the project starts, it's behind—never does get caught up, gets a bad reputation, and lots of times fails, simply because they had no idea going in what the estimate should have been.

**CROSSTALK:** Are we attacking the "same old problem?"

**Randy:** I think the problems are multi-faceted—it's a horrible word to use—but we have two things occurring. One is within the military and DoD projects: People move into the estimating profession, they're there for three years, then they go onto another profession because there is no future and really no promotions possible as an estimator. Then there are those who only develop an estimate once every four or five years, so they really never develop the skill. What we get are a lot of people who pick up an estimating tool, look at the instruction manual for 15 minutes, punch some numbers into the computer, and the computer tells them something that they interpret as truth—that is the estimate that people typically use. Real professional estimators, the people who master and use the skill for a long period of time, are very good and worth seeking out or utilizing.

**CROSSTALK:** With that said, is there still a software crisis, and would this be the crux of the problem as you see it?

**Randy:** Yes. The software crisis included problems like we couldn't manage the costs, we couldn't estimate the schedule, we couldn't maintain the software, and we couldn't modify the software. There were eight characteristics of software that caused the term "software engineering" to even happen. And the experts thought by changing the name that would make the problems go away, but the managers didn't go away, and the problems didn't go away either. We still can't maintain the software, the software still has errors, and the software is still not delivered on time. We've tried.

The first really big effort I saw to get the problems under control was in the late '80s with the CMM®. We thought if we got the processes right, then the productivity would be better, and we would make fewer errors. We tried to fix the problems with structured programming, and then structured design, and then structured analysis.<sup>1</sup> But we've heard the same mantra over and over: *Each of these things will cause errors to disappear and productivity to improve by an order of magnitude*. What we have now is an organized process doing things in an orderly way—and producing the same stuff we did before. I always refer back to a phrase that said, "When processes are optimized, people are interchangeable." If we get the process right, it doesn't really matter who the people are. But it's the people who are the problem. It's also the people who are the answer.

**CROSSTALK:** What I know of tools and your teachings is that you factor in that people issue quite heavily. What is the people issue, as you see it?

**Randy:** I put it into two categories. The first category comes down to their ability to communicate with each other. If we go back to the old Skunk Works® that Lockheed built up millions of years ago, they had a tremendously high productivity of turning out good products in a relatively short period of time, and it worked. The scheme was to give everybody on the project access to everybody else. If you had a question you could go ask somebody a question. Oddly enough, the first cubicle was introduced by the Skunk Works, only it was a mobile workstation that they could move around so they could be where they needed to be when they needed to solve problems.

Someone else found they could maximize the number of programmers per outlet if they put all the cubicles side-by-side. And we have little walls and places where people could work without interfering with anybody else—they could work by themselves. This was exactly the wrong thing to do, I think, from that point of view.

Managers in general look at programming the same way their mentors taught them: You sit down, you code, you work by yourself, and you are in a very productive environment—which is exactly false. We learn in kindergarten, grade school, and high school that if you work with somebody else you are copying and cheating. So we are taught as children to work alone. We

don't learn about working with somebody else until we get older, and, by then, habits are developed and people think they need to work alone—and that's not really true. Managers come up and follow the same idea: If people are talking to someone else, they are wasting time. If people go to the coffee machine, they are wasting time. They should be working, and that's what we're paid for. So if we work on a problem and can't solve it, even though it may take three or four weeks, we *will* finally ask someone for help. But that interaction should be going on *all the time*.

One of the keys of the agile movement—not that all the keys of the agile movement are good ones—is that people are more important than process. If managers recognize the need for their teams to work together, they will. Software Skunk Works are not entirely unique. They've been tried and tested, and they pay off very well. The management response to those organizations is typically, "That's not the way we do business."

Software is not accounting. You're not looking at a column of numbers and trying to make them balance. It's not like almost any other activity I can think of. It's very creative. Every piece of software you write is new and unique, and sharing that development is one way to really improve both productivity and experience of the people who are involved in the communication. It makes the whole organization better.

**CROSSTALK:** I've heard you state that it's good that everyone understands software cost estimation, but who specifically on a software project team really needs to know the ins and outs of cost estimation?

**Randy:** The estimator, who should be an expert with the tool. There are half a dozen tools that estimate projects in an equal number of ways. They approach problems differently, but those estimators who are good at using those tools will get nearly the same answers using different approaches if they understand what they are doing. You don't hand an accountant a scalpel. There has to be somebody in the organization who really knows how to estimate.

On the other hand, it won't hurt the manager to take an estimating class and understand the meanings of the parameters. By understanding the meanings, they can understand quantitatively what the impact of their decisions will be. If I force people into cubicles, my productivity is going to go down about 20%. If you count the analyst as well that's 40% that we've lost. If you give the team tools that don't work, that's another 10 to 12%. If you convert your organization into a well-functioning Theory-Y<sup>2</sup> Skunk Works, you might double or triple productivity. You can look at the individual things that can contribute to cost, and you say, "Yeah, that makes sense, it will work."

I have one good example from my prior life. The project manager took an estimating class. He learned the tool and he did the homework. When he started the next project, he said, "I'm going to do everything the estimating model tells me to do." He located an unused cafeteria as a working Skunk Works. His programming team cleaned up the tables and the floors and he brought in a

microwave (he said he bought a lot of popcorn). His task was to keep other people out of the way, which he did. The manager's role was to support the team and keep them moving forward with the development. What he gained was the highest technology rating (I have a numerical way of rating technology)<sup>3</sup> I have ever seen, as well as achieving productivity that had never been seen by that organization. It was about 150% better than the norm.

**CROSSTALK:** DoD is supporting a massive effort to obtain data leading to a new estimating tool. You've looked at data, what are your thoughts on quality of data?

**Randy:** I've spent almost a year analyzing a total of 960 some-odd data points that had been stored in a DoD software product database. At the end of the year, I found about 15 data points that matched what I would consider reality.

The analysis showed that when you included all the data, the development effort was independent of size. When we filtered the data (down to the final 15 points), we realized the "data base" was actually a data repository, not the database as advertised. Most databases are, in reality, repositories. The two should never be confused. We turned in a report and we haven't heard another word from them. I don't expect we will. I don't think the analysis was what they wanted to hear.

There is a real concern with data quality: We can't seem to get our hands around cost and schedule data. There's the CHAOS report<sup>4</sup> showing that about a third of projects are never completed within a reasonable schedule and cost. And that problem will go on and on.

**CROSSTALK:** Do you think the industry has warmed up to the idea of the people factor? Do you think it is generally accepted?

**Randy:** No. Again, it's not the way they do business, or should I say, it's outside their organization culture. I've been studying this for about 30 years. If you look at publications on the people issues versus the tool and process issues, you'll see the vast majority focus on technology while only a few look at people issues.

I remember one collaborative project where the error rate was three orders of magnitude less than normal and productivity was much higher than that organization had ever done before. When it was presented to their management staff, one of the project managers said, "If we forced our senior people to work with someone else, our senior people would all quit." Now, I looked at the numbers (sitting in the back of the room), and said, "You know, if all of their senior people quit, the productivity would increase."

Anyway, the company was determined they couldn't do it because they thought that people don't like to work together—and I think that's absolutely wrong. I've never seen that in a team-oriented environment.

**CROSSTALK:** Along those same lines then (these people issues), are we able to accurately estimate and measure human components?

But “socialize” is exactly what they want to do. Even when you’re “wasting” time, you’re communicating, and in wasted time there are usually work problems that come into it—you end up talking about what you’re stuck on.

**Randy:** Yes we can. It's not a hard science, mind you. It is like physics in some ways. The major estimating models dating back to 1980 have all predicted the effect of the human components.

We have mapped out a group of very strong indicators of what we call a capability rating. Capability ratings are not just I.Q. and programming skill and experience: It includes motivation, management style, and the ability to communicate and to work with others—your personality issues and the ability to talk to somebody. To take one from contemporary lore, Sheldon on *The Big Bang Theory*<sup>5</sup> is a perfect example of one of those people who would be very low on the capability rating: In spite of being terribly intelligent, he can't communicate that intelligence to anybody else. And that's important.

So one of the first questions I ask myself when looking at an organization is, “Are they talking?” The first test that I use when I walk into a work area—and look over the sea of cubicles—is the noise level. What you hear makes all the difference in the world.

One in particular is a perfect example: They had the world's best-looking cubicle environment I had ever seen—all uniform, all had the same equipment, same manufacturer, and everybody had two displays to work with. The organization was set up so that the people who built the previous system, the old-timers, were all on one side of the room; the people building the new system were on the other side of the room. In talking with the older group, they told me that they had all of this experience they were transferring to this younger group. When we walked in the room, I stopped and said, “Just simply listen.” We listened for a few minutes, and there was not a sound. I said, “Who's communicating in here ... anybody?” The answer was, “Well, they're all doing it online—the Internet, IMs, e-mails, what have you.” And walking through, programmers were at their workstations, quietly at their computers entering things. I told them that the *content* of the communication between people—in their case between the experts and the novices—(according to the study I referred to) is only about 7%<sup>6</sup> of the communication. So a lot is lacking without face-to-face discussion. They may as well be reading documents because the interaction they're getting is slow, unclear, and you can't quickly discuss it or argue about it. So, yes, that's an issue.

Another thing we ask is, “Are the cubicles big enough that somebody can come into your cubicle, sit down, and talk about a problem?” In this organization there was not. If they came to ask a question, they leaned over the wall—and there was no one

doing that. You could have areas with conference tables and lots of white boards where they could discuss problems—and this organization didn't have any of those either.

One of the last things we looked for—and it turns out to be one of the most significant, I don't know why, it just works that way—is the smell of popcorn. Popcorn indicates that people are talking—you don't sit and eat popcorn by yourself. We had this conversation with one of the younger employees:

**If you get stuck on a problem, who do you go to?**

**Well, I can send an e-mail to so-and-so.**

**Is there anyone you can talk to?**

**Well, we're not supposed to talk. We're not supposed to socialize.**

But “socialize” is exactly what they *want* to do. Even when you're “wasting” time, you're communicating, and in wasted time there are usually working problems that come into it—you end up talking about what you're stuck on.

**CROSSTALK:** So it's a hard science, but measurable nonetheless.

**Randy:** It's all communication. The whole thing. And it's very easy.

**CROSSTALK:** It seems, at least through your experience, that many organizations seem to be having a hard time not just grasping the need for interaction but the actual quality of the physical environment in which employees work.

**Randy:** I went to this company that was having trouble delivering a system. Hypothetically, they were in Bozeman, Montana: cold, windy, and just plain miserable. We went to their engineering building, which was a recovered, retrofitted livery stable. Picture one of those old stables: big double doors where wagons went in and out, a stone building. There were rows of tables in there where they were assembling the components for the systems they were delivering. I asked, “Where are the software people? I don't see them anywhere.” I was led through a door in the back into a lean-to on the back of the building. It had a corrugated steel roof and tiny factory windows—some of which were broken out. Inside there were rows of tables with people huddled over their terminals, blazing away at this software: They were all wearing overcoats, typing with gloves, no Internet, with the only heat being generated by their computers. And, as a capstone to this whole thing, the floor was *dirt*. I said, “This is insane, why are you here? I can't believe this.” They said, “There's no place within 50 miles where we can find work. This is the place—if you want to program, you work here.” They were all there, not delivering anything, freezing to death. I explained the observation to the vice presidents the next day saying, “These people hate this place.” They said “Why? We're giving them all this opportunity.” They could not understand that environment and motivation had something to do with their lack of success.

**CROSSTALK:** One final question. If I'm a software project manager and I'm reading this interview, what are some of the things I can do today to make me better at cost estimation?

**Randy:** That's a real tough one. No one has asked me that question before.

One, I would review a cost estimating user manual and see what parameters are used in the estimate—focus in on the parameters that have the largest effect. Spend a whole day just reading the manual and understanding the cost impact of the decisions you're going to make on a project. That would be very worthwhile.

I think Jerry Weinberg said it best—he has a law that says that everybody has a problem, and it's always a people problem.<sup>7</sup> And most projects I've looked at bear that out.

I could go on all day talking about this.♦

*Disclaimer:*

<sup>®</sup> *CMM is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.*

<sup>®</sup> *Skunk Works is registered in the U.S. Patent and Trademark Office by Lockheed Martin.*

## NOTES

1. See <[http://en.wikipedia.org/wiki/Structured\\_analysis](http://en.wikipedia.org/wiki/Structured_analysis)>.
2. See <[http://en.wikipedia.org/wiki/Theory\\_X\\_and\\_theory\\_Y](http://en.wikipedia.org/wiki/Theory_X_and_theory_Y)>.
3. For more on rating technology, see Jensen's co-authored CrossTalk article (with Lawrence H. Putnam Sr., and William Roetzheim) at <<http://www.crosstalkonline.org/storage/issue-archives/2006/200602/200602-Jensen.pdf>>.
4. See <[http://www1.standishgroup.com/newsroom/chaos\\_2009.php](http://www1.standishgroup.com/newsroom/chaos_2009.php)>.
5. A sitcom on CBS. See <[http://en.wikipedia.org/wiki/Sheldon\\_Cooper#Characteristics](http://en.wikipedia.org/wiki/Sheldon_Cooper#Characteristics)>.
6. With 38 percent being tone of voice and 55 percent body language, according to Albert Mehrabian's 7-38-55 Rule. See <[http://en.wikipedia.org/wiki/Albert\\_Mehrabian](http://en.wikipedia.org/wiki/Albert_Mehrabian)>.
7. From Gerald W. Weinberg's *Secrets of Consulting: A Guide to Giving and Getting Advice Successfully*, specifically the First and Second Laws of Consulting (pg. 5).



## AWARD WINNERS

# DoD Systems Engineering Top 5 Program Awards

**Sponsored by Department of Defense Systems Engineering Directorate and National Defense Industrial Association Systems Engineering Division**

The awards, presented to both government and industry, recognize significant systems engineering achievement by teams of industry and government personnel.

**Winners:**

- Army: Advanced Field Artillery Tactical Data System (AFATDS)
- USAF: Battlefield Airborne Communications Node (BACN) JUON
- Army: Base Expeditionary Target & Surveillance Systems-Combined (BETSS-C)
- Army: Defense Readiness Reporting System-Army (DRRS-A)
- USAF: C-17 Globemaster III Modernization

Awards presented at the annual NDIA Systems Engineering Conference  
San Diego, CA, October 25–28, 2010

<http://www.acq.osd.mil/se/apr/top5awards.html>



# Homeland Security

The Department of Homeland Security, Office of Cybersecurity and Communications, is seeking dynamic individuals to fill several positions in the areas of software assurance, information technology, network engineering, telecommunications, electrical engineering, program management and analysis, budget and finance, research and development, and public affairs. These positions are located in the Washington, DC metropolitan area.

To learn more about the DHS Office of Cybersecurity and Communications and to find out how to apply for a vacant position, please go to USAJOBS at [www.usajobs.gov](http://www.usajobs.gov) or visit us at [www.DHS.GOV](http://www.DHS.GOV); follow the link Find Career Opportunities, and then select Cybersecurity under Featured Mission Areas.

# Data Mining for Process Improvement

Paul Below, Quantitative Software Management, Inc. (QSM)

## Introduction

What do you do if you want to create an estimate and you have 100 candidate variables to use in your estimating model?

This is also a common question for CMMI® high maturity organizations that need to create process performance models. According to SEI, process performance models are:

"A description of relationships among attributes of a process and its work products that is developed from historical process-performance data and calibrated using collected process and product or service measures from the project and that are used to predict results by following a process."

High maturity organizations typically use process performance models for operational purposes such as project monitoring, project planning, and to identify and evaluate improvement opportunities. They typically are used to predict many output variables including defects, test effectiveness, cost, schedule and duration, requirements volatility, customer satisfaction, and work product size.<sup>1</sup>

Data mining techniques can be used to filter many variables to a vital few to build or improve predictive models. Specific examples are provided in four categories: classification, regression, clustering, and association.

When creating an estimating model or a process performance model, the primary challenge is how to start. Regardless of the variable being estimated (e.g., effort, cost, duration, quality, staff, productivity, risk, size), there are many factors that influence the actual value and many more that could be influential.

The existence of one or more large datasets of historical data could be viewed as both a blessing and a curse. The existence and accessibility of the data is necessary for prediction, but traditional analysis techniques do not provide us with optimum methods for identifying key independent (predictor) variables from a large pool of variables.

Data mining techniques can be used to help thin out the forest so that we can examine the important trees. Hopefully, this article will encourage you to learn more about data mining, try some of the techniques on your own data, and see if you can identify some key factors that you can control or use to build a predictive model.

## What Is Data Mining?

There are many books on data mining, and each one has a slightly different definition. The definitions commonly refer to the exploration of very large databases through the use of specialized tools and a process. The purpose of the data mining is to extract useful knowledge from the data, and to put that knowledge to beneficial use.

Data mining can be viewed as an extension of statistical analysis techniques used for exploratory analysis, incorporating new techniques and increased computer power. A few sources are listed in the resources section that provide details on data mining.

There are a number of myths that have grown up regarding the use of data mining techniques. Data mining is useful but not a magic box that spits out solutions to problems no one knew existed. Still required for success:

- business domain knowledge
- the collection and preparation of good data
- data analysis skills
- the right questions to ask

Techniques for cleansing data, measuring the quality of data, and dealing with missing data are topics that are outside the scope of this article.

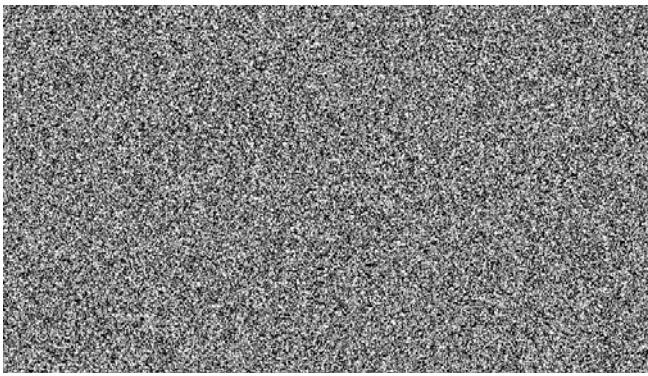
Researchers have created a number of new data mining algorithms and tools in recent years, and each has theoretical advantages and avid proponents. However, for the purpose of getting started with estimate model creation, tool selection is not critical. The comparative theoretical advantages and disadvantages of the techniques and tools is not important to our purpose of identification of key factors. The practical advice is to try as many different techniques as possible, as the difficult time-consuming task is data preparation. Refer to a list of tools in the References section.

## Model Creation Challenges

People love to interpret noise. Regardless of what the data shows, the audience will offer theories to explain the causes for what is observed. If a graph shows that performance has improved, someone will offer an explanation for why that happened. If you tell the audience that the graph was upside down, and performance has actually decreased, just as quickly someone will propose a reason for why that happened.

Figure 1 is an image of random noise. If you stare at it long enough, you will start to see some patterns. People are pretty good at pattern recognition, even if no pattern actually exists. That is one reason why statistical quality control, data mining, and hypothesis testing are useful—to help us see whether the patterns we think we see are real or whether they could be explained by randomness alone. Another reason is to help us find patterns that are real but are difficult to see.

Figure 1: Random Noise



Exploratory analysis, including data mining, utilizes existing data that has already been collected. There are challenges with using such data, including:

- The databases already exist and almost always were created without considering analytical needs.
- Databases generally are built by committees, or have evolved from older systems through multiple stages. The variables stored include items that were used long ago as well as fields that someone thought might be useful someday, mixed in with data that are currently necessary. Many of the fields have values that are hard to decipher, or were used inconsistently by different populations of users.
- The structure of the data is often bad or the keys are not appropriate, making data extraction difficult.

Regardless of the data mining tools used, data extraction and validation is a major undertaking.

Once the data is extracted and placed in a readable format, the analyst is faced with dozens of input variables. Which of those variables should be used in the model?

It is common for our variables to exhibit collinearity. Collinearity is when the variables are highly correlated with each other. In practical terms this means that those variables are measuring the same or similar things. Dumping all of these variables into a regression equation is not a way to receive a useful output.

Data mining can help us thin out the forest so that we can see the most important trees. Many of the data mining techniques can be used to identify independent variables that are influential in predicting the desired result variable. Success will depend more on the mining process than on the specific tools used.

## Data Mining Models

**“Statisticians, like artists, have the bad habit of falling in love with their models.”**

- George Box

Data mining can aid in hypothesis testing as well as exploratory analysis.

There are many pure data mining products on the market, but they are typically very expensive. Some of the common techniques, however, are supported by basic statistical analysis tools which are much less costly. These techniques include all of the examples provided in this document. Examples of statistical analysis tools that support some or all of these functions are listed in the References section.

Data mining models can be placed into four categories as described in this table:

Table 1: Data Mining Models

Category	Description	Purpose	Primary Data Type
Classification	Split the data to form homogenous subsets	Predict response variable	Discrete is best
Regression	Best fit to estimating model	Predict response variable	Continuous (ratio or interval)
Clustering	Group cases that are similar based on selected variables	Identify homogeneous groups of cases	Any
Association	Group variables that are similar	Determine co-linearity, identify factors that explain correlations	Ratio or interval (not categorical)

We will now look at an example from each of the four categories.

## Classification Example

One classification technique is a tree. In a tree, the data mining tool begins with a pool of all cases and then gradually divides and subdivides them based on selected variables.

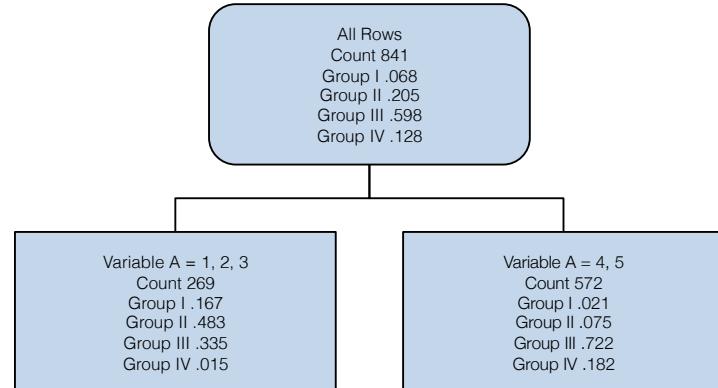
The tool can continue branching and branching until each subgroup contains very few (maybe as few as one) cases. This is called overfitting, and the solution to this problem is to stop the tool before it goes that far.

For our purposes, the tree is used to identify the key variables. In other words, which variables does the algorithm select first? Which does it pick second or third? These are good candidate variables to be used in an estimating model, since the tree selected them as the major factors.

In Table 2, we see an example that started with a data set of 841 cases, taken from a database of client information. Prior to running the tree, each of the 841 clients was assigned to one of four groups. The assignments were made based on information about customer satisfaction. The goal of the analysis was to see if there were key factors that could be used to predict which group a client would fall in. This prediction would then be used to identify clients that were likely to become less satisfied in the future, and determine actions that could be taken to improve client satisfaction.

In the top box of the tree, each group is listed with the fraction of the cases. So, for example, Group I contains 6.8% of the 841 cases. The total for the four groups will be approximately equal to 1 (100%) allowing for round off.

Table 2: Tree Example



The tree algorithm examined all of the variables and selected Variable A to be the first branch. Variable A has possible integer values from 1 to 5. As we can see, the algorithm put the cases where Variable A is equal to 1, 2, or 3 in the left branch and those with Variable A equal to 4 or 5 in the right branch.

The left branch has 269 cases, including most of the cases in Groups I and II (the 269 cases are composed of 16.7% Group I and 48.3% Group II, compared to the right branch which is composed of only 2.1% Group I and 7.5% Group II). The right branch ended up with 572 cases, including most of the cases in Groups III and IV.

Variable A by itself is not a sufficient predictor to use as a predictive model. However, the tree is telling us that Variable A is one important factor.

The tree would have additional branches, but Table 2 is sufficient to aid in explaining how the tree is used.

### Regression and Correlation Examples

The data used in the remaining examples came from industry data. It is based on a sample of 193 projects extracted from a corporate database.

The output in the examples is for illustrative purposes and should not be used to reach conclusions about performance of specific software projects.

Stepwise regression is a type of multivariate regression in which variables are entered into the model one by one, and meanwhile variables are tested for removal. It can be a good model to use when supposedly independent variables are correlated. Stepwise regression is one of the techniques that can help thin out the forest and find important predictive factors.

Table 3 is a summary output of a stepwise regression that went through nine steps to build the best model. It was created in SPSS, although other statistical packages produce similar results. The dependent variable being predicted was errors detected prior to deployment. The stepwise regression selected nine variables that fit the threshold for inclusion, while excluding 20 other variables (not listed).

Table 3: Regression Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
9	.840	.706	.691	330.332

Model		Sum of Squares	df	Mean Square	F	Sig.
9	Regression	47883321.563	9	5320369.063	48.757	.000
	Residual	19968796.365	183	109119.106		
	Total	67852117.927	192			

Predictors: (Constant), Effective SLOC, Life Duration (Months), MB Time Overrun %, MB Effort (MM), Life Peak Staff (People), Data Complexity, MBI, MB Effort %, Mgmt Eff.

Dependent Variable: Errors (SysInt-Del)

The nine variables selected by the stepwise regression were, in the order the tool selected them: effective source lines of code; project life cycle duration in months; percent of duration overrun of Main Build (design through deploy); Main Build man months of effort; peak staff; data complexity; Putnam's Man-power Buildup Index; percent of effort expended in Main Build;

and management effectiveness. Note that two of these nine variables (data complexity and management effectiveness) are qualitative, scored on a scale of one to 10 where five is average and 10 is high.

The first number to look at in Table 3 is the Sig (significance) in the rightmost column. The most commonly used significance threshold is .05, which means that the variable or model would be significant at the 95% level. In the example, the value .000 means that we have less than a one in a thousand chance of being fooled by random variation into thinking this model is significant.

Although all nine variables selected are clearly significant, the overall model created has an adjusted R square of .691, which means that these nine variables taken together are explaining about 69% of the variation in errors found. This may not be the best model to use for estimating, but it is important to look at each of the nine variables if the intent is to create an estimating model or if we need to reduce the number of errors found in the future.

The coefficients of the stepwise regression formula are displayed in Table 4. Each variable is listed next to the coefficient B, which is the multiplier in the linear equation.

Table 4: Regression Coefficients

Coefficients:					
Dependent Variable: Errors (SysInt-Del)					
Variable	Unstandardized Coefficients		Sig.	95% Confidence Interval for B	
(Constant)	-580.411	239.656	.016	-1053.255	-107.568
Effective SLOC	.001	.000	.000	.001	.001
Life Duration (Months)	27.633	5.832	.000	16.126	39.139
MB Time Overrun %	.026	.006	.000	.015	.037
MB Effort (MM)	1.535	.326	.000	.892	2.177
Life Peak Staff (People)	-.7438	1.905	.000	-11.197	-3.679
Data Complexity	66.840	18.269	.000	30.795	102.886
MBI	33.683	14.609	.022	4.859	62.507
MB Effort %	3.924	1.552	.012	.862	6.987
Mgmt Eff.	-50.012	22.775	.029	-94.948	-5.076

The equation that yielded the adjusted R square of .691 is:

$$\text{Errors} = -580 + (.001 * \text{ESLOC}) + (27.6 * \text{Duration}) + (.026 * \text{overrun\%}) + (1.5 * \text{MB Effort}) - (.74 * \text{peak staff}) + (66 * \text{data complexity}) + (33.68 * \text{MBI}) + (3.9 * \text{MB effort \%}) - (50 * \text{Mgmt Eff})$$

The factors in the equation can be determined from reading the numbers in the B column.

A negative number means a negative correlation. One counterintuitive result of this example is the coefficient for peak staff. The negative coefficient means in this model the larger the peak staff the smaller the number of errors detected. This type of result is why it is necessary to evaluate the data in more depth and do additional analysis before using the model. Sometimes, negative correlations are expected. For example, management effectiveness has a negative coefficient meaning that a higher effectiveness results in a lower number of errors.

The two rightmost columns, the 95% confidence intervals, are useful as an indication of the uncertainty in the coefficients. The lower and upper bound for any variable should not straddle zero. If it did, that would be an indication that we lack confidence in the factor B. Another method is to compare the value of the standard error to the value of the coefficient; ideally the standard

error should be much smaller than the coefficient B. Also, the Sig should be small, ideally less than .05.

In addition to regression, correlation can be used to identify candidate important variables. This can be done by selecting the dependent variable first for the correlation and then the list of independent variables. There are different types of correlation that can be used. For ratio data, Pearson correlation can be used. For ordinal data, Kendall's Tau-B will work. For nominal (categorical) data, a chi square test can be used on a crosstab (two-way table) to determine significance.

It is important to note that these tests will determine linear correlations. Sometimes correlations exist but are nonlinear. One technique for exploring those relationships is transformation, which is not discussed in this paper.

### Clustering Example

Cluster techniques detect groupings in the data. We can use this technique as a start on summarization and segmentation of the data for further analysis.

Two common methods for clustering are K-Means and hierarchical. K-Means iteratively moves from an initial set of cluster centers to a final set of centers. Each observation is assigned to the cluster with the nearest mean. Hierarchical clustering finds the pair of objects that most resemble each other, then iteratively adds objects until they are all in one cluster. The results from each stage are typically saved and displayed numerically or graphically as a hierarchy of clusters with subclusters.

Table 5 is the output of a K-Means example run from the sample with the output constrained to create exactly three clusters.

The tool placed the largest projects in the first two clusters. These projects had more errors, more staff, and higher productivity than the third cluster. One difference between the first two clusters is that the projects in the second cluster tended to have poor estimates of effort.

Table 5: Cluster Example

#### Final Cluster Centers

	Cluster		
	1	2	3
Project Count	5	22	166
Life Effort (MM)	750.7	617.8	89.1
Errors (SysInt-Del)	1898	1030	186
Errors First Month	138	117	8
Total FP	37167	26533	2648
Effective SLOC	1272194	298791	26444
Life Duration (Months)	21.3	18.4	9.3
Life Peak Staff (People)	56.5	61.1	15.4
Life Avg Staff (People)	23.8	26.5	7.1
MB Eff Overrun %	.0	62.0	45.8
SLOC/MB MM	2384.5	1606.4	910.9
Putnam's PI	24.4	21.5	14.1

We may want to stratify the projects into groups based on the above distinctions prior to conducting additional analysis. This may result in the need for more than one estimating model, or more than one process improvement project.

### Association Example

Association examines correlations between large numbers of quantitative variables by grouping the variables into factors. Each of the resulting factors can be interpreted by reviewing the meaning of the variables that were assigned to each factor. One benefit of association is that many variables can be summarized by just a few factors.

In the following example using actual data, Principal Components analysis was used to extract four components. The Scree Plot in Figure 2 was used to determine the number of components to use. The higher the Eigenvalue, the more important the component is in explaining the associations. Selection of the number of components to use is somewhat arbitrary, but should be a point at which the Eigenvalues decline steeply (such as between components 2 and 3, or between 4 and 5). It turned out in this example that the first four components account for roughly half of the variation in the data set (included in other output from the principal components tool, not shown here), making four a reasonable choice.

Figure 2: Scree Plot for Association example

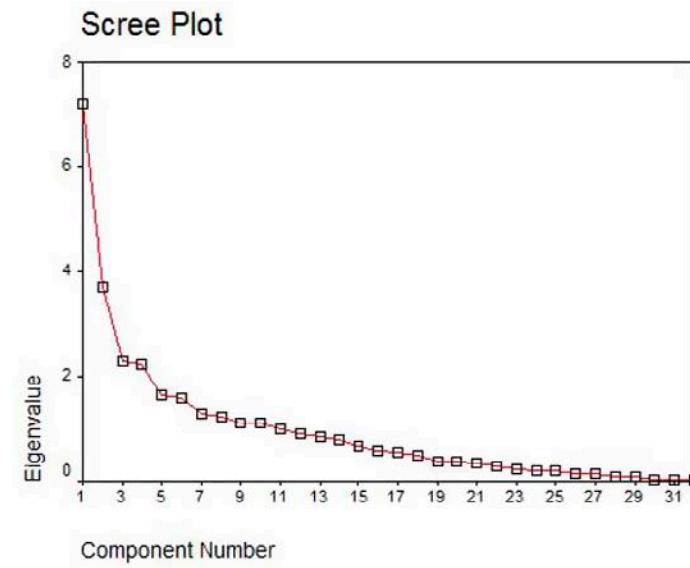


Table 6 displays variables with the most significant output for each component. The important numbers in the table are those with relatively large absolute values and have been shaded for easy reference.

- Component 1 is composed of a market basket of variables related to effort and size (the variables aligned with the shaded numbers in component 1).
- Component 2 grouped variables related to the development team: knowledge, turnover, and skill.
- Component 3 isolated the Manpower Buildup Index, which is the speed at which staff is added to a project.
- Component 4 linked the percent of effort expended in functional requirements to the percent expended in the Main Build (design through deploy).

Variables that are seen to be related should be combined (or one should be chosen as the representative) as an input variable when creating prediction models or identifying root causes.

*Table 6: Association example output*

	Component			
	1	2	3	4
Life Effort (MM)	.920	-.152	.196	-.006
Effective SLOC	.652	.111	-.475	.106
Life Duration (Months)	.658	-.198	-.429	.066
Life Peak Staff (People)	.865	-.115	.338	-.137
Life Avg Staff (People)	.823	-.157	.381	-.156
FUNC Effort (MM)	.880	-.169	.151	.098
MB Effort (MM)	.925	-.160	.065	-.122
Func Effort %	-.241	.088	.236	.719
MB Effort %	-.059	-.072	-.247	-.765
Knowledge	.186	.770	.161	-.076
Staff Turnover	.083	-.717	.049	.110
Dev Team Skill	.133	.746	.029	-.225
MBI	-.006	-.011	.640	-.200

## Summary

Once data has been collected and validated, the hardest work is behind you. Any data mining tools that are available to the researcher can be used relatively quickly on clean data. These data mining techniques should be used to filter an overwhelming set of many variables down to a vital few predictors of a key output (for example, quality).

Determination of the vital few is a key component of process improvement (such as Six Sigma projects) activities as well as prediction. With those key drivers or influencers of quality in hand, improvements can be designed and implemented with fewer iterations, effort, or time.

In addition to process improvement activities, we use the “vital few” to build error prediction models, and then use the models to tune parametric project estimates for specific clients. The project estimate and plan is thereby not only an estimate of duration and cost to complete construction, but also includes the prediction of when the system will be ready for prime time.♦

### *Disclaimer:*

<sup>®</sup> CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

## ABOUT THE AUTHOR



**Paul Below** has over 25 years experience in the subjects of measurement technology, statistical analysis, estimating and forecasting, Lean Six Sigma, and data mining. He has provided innovative engineering solutions as well as instruction and mentoring internationally in support of multiple industries. He serves as services consultant for Quantitative Software Management (QSM) where he provides clients with statistical analysis of operational performance, helping strengthen competitive position through process improvement and predictability.

Paul is a Certified Software Quality Analyst, and a past Certified Function Point Specialist. He is a Six Sigma Black Belt. He has been a course developer and instructor for Estimating, Lean Six Sigma, Metrics Analysis, Function Point Analysis, as well as statistics in the Masters of Software Engineering program at Seattle University. He is a member of the IEEE Computer Society, the American Statistical Association, the American Society for Quality, the Seattle Area Software Quality Assurance Group and has served on the Management Reporting Committee of the International Function Points User Group as well as CMMI high maturity assessment teams. He has one US patent and two patents pending.

**paul.below@qsm.com**  
**Quantitative Software Management, Inc. (QSM)**  
**<http://www.qsm.com/>**

## RESOURCES

### Data Mining Websites:

<<http://www.twocrows.com>> <<http://www.kdnuggets.com>> <<http://www.datamininglab.com>>

### Data Mining Tools:

- Statistical tools that each do some data mining techniques include: SPSS; SAS; JMP; SPlus; Minitab
- Specialized data mining tools include Salford Systems CART and MARS; SAS Enterprise Miner; PASW Modeler (SPSS); Insightful Miner (SPlus)

### Books:

- *Introduction to Data Mining*, by Pang-Ning Tan, et al, Addison-Wesley, 2006.
- *Principles of Data Mining*, by David Hand, Keikki Mannila and Padhraic Smyth, MIT Press, 2001.
- *Data Mining – Concepts, Models, Methods and Algorithms*, by Mehmed Kantardzic, John Wiley and Sons, 2003.
- *Data Mining: Opportunities and Challenges*, by John Wang, IDEA Group, 2003.
- *Use and Organizational Effects of Measurement and Analysis in High Maturity Organizations: Results from the 2008 SEI State of Measurement and Analysis Practice Surveys*, by Dennis Goldenson, James McCurley, Robert W. Stoddard, CMU/SEI-2008-TR-024.

## NOTES

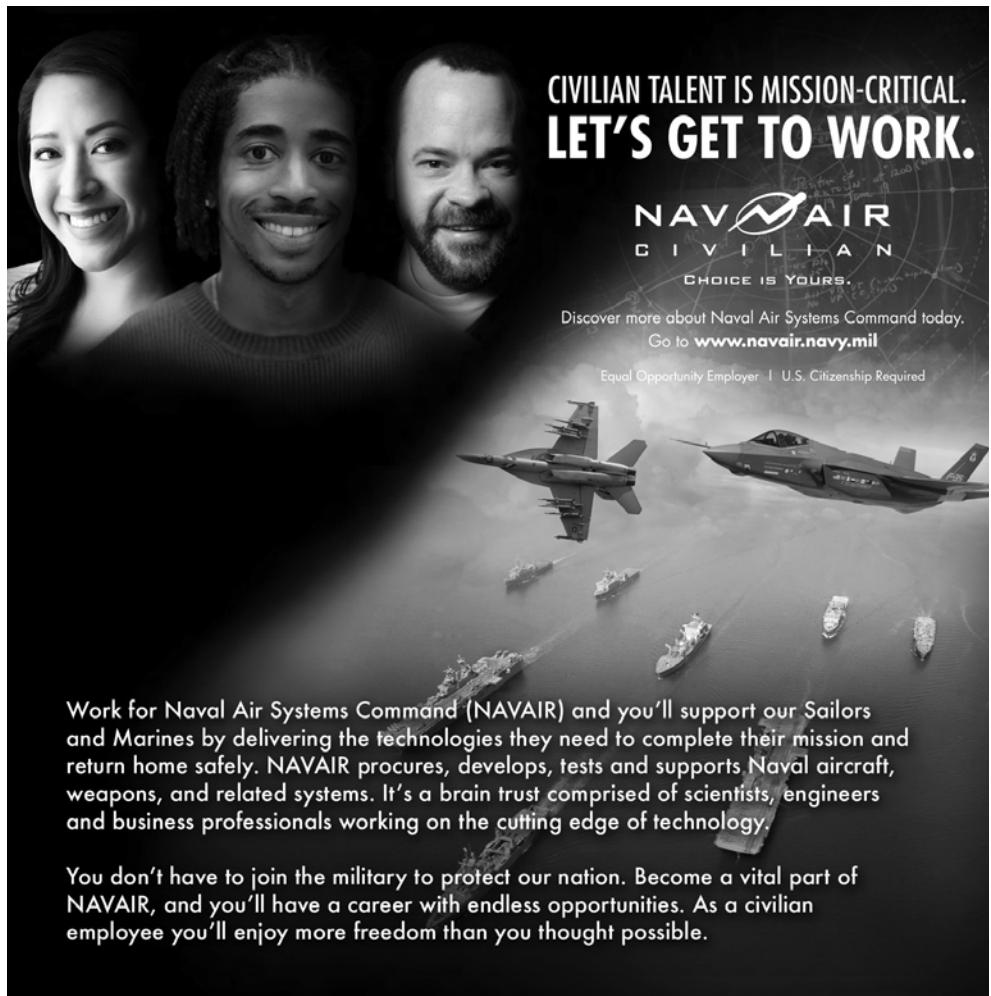
1. CMU/SEI-2008-TR-024

**CIVILIAN TALENT IS MISSION-CRITICAL.  
LET'S GET TO WORK.**

**NAVAIR**  
CIVILIAN  
CHOICE IS YOURS.

Discover more about Naval Air Systems Command today.  
Go to [www.navair.navy.mil](http://www.navair.navy.mil)

Equal Opportunity Employer | U.S. Citizenship Required



Work for Naval Air Systems Command (NAVAIR) and you'll support our Sailors and Marines by delivering the technologies they need to complete their mission and return home safely. NAVAIR procures, develops, tests and supports Naval aircraft, weapons, and related systems. It's a brain trust comprised of scientists, engineers and business professionals working on the cutting edge of technology.

You don't have to join the military to protect our nation. Become a vital part of NAVAIR, and you'll have a career with endless opportunities. As a civilian employee you'll enjoy more freedom than you thought possible.



# Demystifying Cloud Computing

**Qusay F. Hassan, Faculty of Computers and Information, Mansoura University, Egypt**

**Abstract.** Cloud computing is a new terminology that was added to IT jargon in early 2007. Still, people overuse this idiom to refer to things that may not relate to its actual definition and scope. Is it all about web hosting? Is it an old thing in new clothes? Why should organizations consider it? IT, business, and academia folks ask about cloud computing with the intent to understand it better. This paper tries to demystify cloud computing by simplifying its terms to readers with different IT interests.

## Introduction

Over the last three years, many IT professionals, business managers, and researchers have started to talk about a new phenomenon called cloud computing. Each of these groups defined cloud computing differently according to their understanding of its offerings [1]. Although there was no agreement about what precisely constituted cloud computing, it still offered a promising paradigm that could enable businesses to face market volatility in an agile and cost-efficient manner.

Recently, a concise definition of cloud computing has emerged: To outsource IT activities to one or more third parties that have rich pools of resources to meet organization needs easily and efficiently [2]. These needs may include hardware components, networking, storage, and software systems and applications. In addition, they may include infrastructure items such as physical space, cooling equipments, electricity, fire fighting systems, and human resources to maintain all those items.

In this model, users are billed for their usage of remote IT infrastructures rather than buying, installing, and managing them

inside their own datacenters. This structure gives users the flexibility to scale up and down in response to market fluctuations. For instance, a business enters the market with a new website that is initially unknown to customers, but eventually becomes popular with hundreds of thousands of requests per day. With cloud computing, businesses may start with a minimum set of IT resources and allocate additional services during peak times. Moreover, website owners can easily dispose unused IT resources during non-peak/recession times enabling them to reduce overall costs.

Typically, cloud computing adopts the concept of utility computing to give users on-demand access to computing resources in a very similar way to accessing traditional public utilities such as electricity, water, and natural gas. In this framework, clients follow a pay-as-you-go model that provides access to as much or as little computing resources as needed whenever needed from anywhere. Hence, organizations are no longer obliged to plan ahead and highly invest in computing resources to accomplish business goals.

## History of Cloud Computing

The idea of cloud computing is not actually new as it goes back several decades. It was pioneered by Professor John McCarthy, a well-known computer scientist who initiated time-sharing in late 1957 on modified IBM 704 and IBM 7090 computers [3]. McCarthy expected that some corporations would be able to sell computing resources through the utility business model. Soon enough, different organizations paid for their use of computing resources (storage, processing, bulk printing, and software packages) available at service bureaus.

Over the past two decades, different implementations tried to leverage similar computing models including:

- **Web Hosting:** This service allows individuals and organizations to host their websites on spaces provided by datacenters of other companies. In web hosting, service providers offer different hosting options to clients. Offerings range from free web hosting for personal uses or shared web in which tens of websites are hosted on the same server, to dedicated servers that give each client his own server with full control over it.

- **Application Service Provider (ASP):** A paradigm where software companies offer applications for remote access by clients through networks for monthly fees [4]. ASP model exempts clients from the burden of buying, installing, and maintaining prepackaged solutions and underlying hardware infrastructures by shifting these tasks to providers.

- **Volunteer Computing:** Many research experiments that depend on high volume computing processes meet their needs by exploiting idle computing resources available through volunteers [5]. This paradigm provides researchers with access to super-computer-like performance in a cost-effective manner.

- **Online File Sharing:** Websites enable Internet users to share their files online. For example, Flickr customers can manage and share their photos over the Internet. In this model, shared files are hosted on public spaces that Internet clients can access whenever and wherever needed.

- **Social Networks:** A variety of websites connect users interested in specific subjects. Examples are YouTube, Wikipedia, Blogger, Facebook, and MySpace. All these networks allow their users to share their ideas and resources such as presentations, videos, games, and small computer applications in an easy and efficient manner.

## Definition and Characteristics

A cloud is an on-demand computing model composed of autonomous, networked IT (hardware and/or software) resources. Service providers offer clouds with predefined quality of service (QoS) terms through the Internet as a set of easy-to-use, scalable, and inexpensive services to interested clients on a subscription basis.

These attributes characterize cloud computing:

- **On-demand Computing Model:** Organizations are no longer required to own their datacenters to cover their IT needs; i.e., they can access giant pools of resources offered by providers in a way similar to accessing public utilities.

• **Autonomous:** Clients are unaware of the technical complexities of offered services. Some of these aspects include used technologies, physical location(s), networks, cooling structures, and number of human resources who manage the services.

• **Predefined QoS:** Cloud providers state QoS terms in their service level agreements to inform clients about expected level of service. QoSs give clients the chance to choose from available providers who can fulfill their technical needs.

• **Internet-based:** The name cloud originally came from the cloud shape that is widely used in the IT field to graphically represent the Internet. It means that all cloud services are hosted beyond client boundaries and delivered over the Internet.

• **Easy-to-use:** Cloud providers offer easy-to-use interfaces that enable clients to make use of their services. These interfaces include both GUI forms for administrators and APIs for developers as well.

• **Scalable:** Clients are not limited with fixed amounts of resources. Rather, they can scale up or down their usage according to fluctuating needs. This goal is accomplished through methods that allow clients to dynamically create, upload, and install their virtual machine images either by code or GUI screens.

• **Inexpensive:** Cloud computing gives small-and-medium-sized enterprises (SMEs) that cannot afford their own datacenters a significantly lower-cost option. This savings results from the fact that resources owned by providers are shared among several clients rather than being solely dedicated to specific client.

• **Subscription-based Model:** Clients subscribe to services they are interested in, and they are billed (usually at the end of the month) according to use.

## Architecture

As illustrated in Figure 1, the architecture of cloud computing is like a pyramid composed of four layers listed from bottom to top as follows [6]:

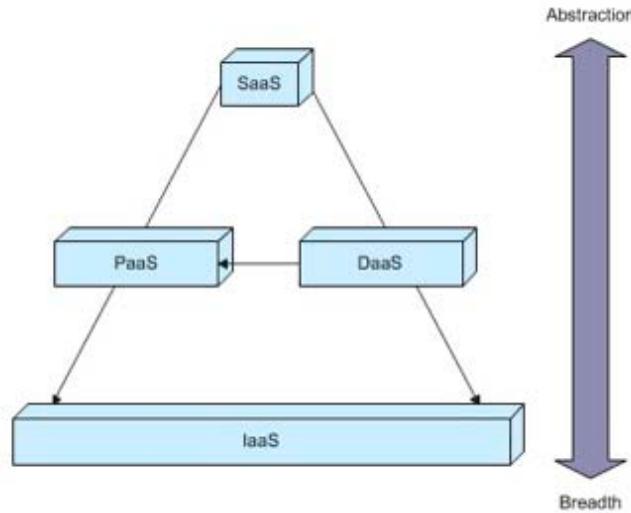
- **Infrastructure-as-a-Service (IaaS):** Represents the base of the pyramid without which the whole architecture cannot exist. IaaS provides hardware such as CPUs, memory, storage, networks, and load-balancers. Examples of IaaS providers include Amazon, Rackspace, and GoGrid.

• **Platform-as-a-Service (PaaS):** Supplies users with development and administration platforms that provide on-demand access to available hardware resources. Many PaaS platforms are available to enable access to IaaS resources. Examples of PaaS platforms include Amazon Web Services, Google App Engine, Windows Azure, and Force.com.

• **Data-as-a-Service (DaaS):** Frees organizations from buying high-cost database engines and mass storage. This service offers database capabilities for storing client information. Examples of DaaS include Amazon Simple DB, Amazon RDS, Google BigTable, and Microsoft SQL Azure Database.

• **Software-as-a-Service (SaaS):** The ultimate form of cloud resources that delivers software applications to clients in terms of accessible services. With SaaS, clients subscribe to applications offered by providers rather than building or buying them. Developers can also enrich their applications by integrating SaaS services into them. SaaS services may be designed to access cloud databases through a DaaS layer, or they may be designed to access hardware resources only through a PaaS layer. Examples of SaaS solutions/providers include Google Apps, Microsoft Online Services, and Salesforce.

Figure 1: Cloud Computing Architecture



## Enabling Technologies

As illustrated in Figure 2, different technologies converged and worked together to enable the emergence of cloud computing, including:

• **Broadband:** High-speed Internet access enabled systems and data to reside in one continent while users access them from different continents. Furthermore, it increases accessibility to large data files such as images, videos, audios, and other

binary and large objects. The ability to do so has given organizations the flexibility required to overcome economic constraints in order to accomplish their goals. For example, many enterprises choose to host their information systems in less expensive datacenters in developing countries in Asia, Africa, and Eastern Europe to reduce their IT costs.

- **Grid Computing:** A distributed computing model that tends to gather underutilized computing resources available in organizations to process computing-intensive tasks faster [7]. This model has given organizations like Amazon the idea to lease unused resources (both processing units and storage) to clients in need of them.

- **ASP:** In the mid 1990s, ASP came to the surface as a business model that enabled organizations to access applications hosted by third parties, freeing them to focus on their business instead of being distracted by IT complexities. In fact, this model was not widely adopted due to two reasons [8]. First, it was unacceptable to organizations that had already invested in complex and expensive systems to reinvest in other new systems. Second, SMEs with no experience of outsourcing did not like to take chances until best practice scenarios were presented by bigger adopters. In this context, it is worth mentioning that although this model failed to survive, it opened the door to organizations outsourcing their applications to third parties.

- **Service-Oriented Architecture (SOA):** The idea of SOA is to turn functionalities of both existing and new applications into a set of granular components [9]. SOA has encouraged software vendors to offer their products as services that clients can use/reuse and compose together to fulfill business requirements in an agile manner. This agility applies to cloud computing as well making it easier to access available hardware and software resources.

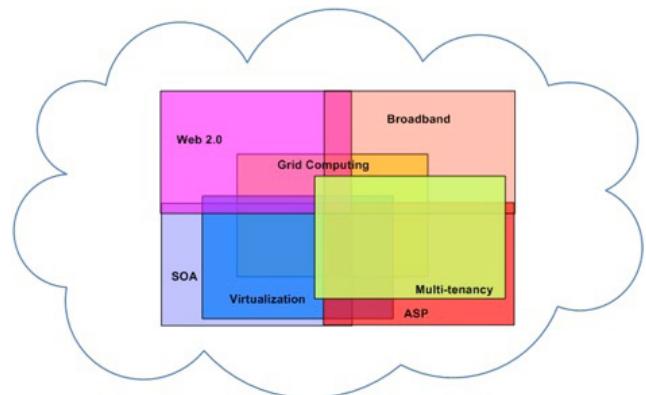
- **Web 2.0:** The last 10 years has seen many advances in web technologies. Innovations included different data formats and forms of accessibility to information available on the Internet such as RSS, Blogs, Portals, Wikis, XML, Web Services, and other mashups [10]. These techniques helped organizations to offer their information as sets of services that allow others to easily access them to mix and match underlying functionalities in their own websites/applications.

- **Multi-tenancy:** A software architecture that allows software vendors to offer a single instance of one or more of their systems to different tenants (clients). Multi-tenancy represents an evolution of the ASP model that offers similar services at lower costs. The savings mainly come from sharing the same software instance and underlying infrastructures by clients rather than dedicating resources to each single one. Technically, this model depends on a single database that stores tenant information with virtual separation between them. The separation is usually made by partitioning data into different sets of records, each of which is marked with the account ID of a corresponding client.

- **Virtualization:** Hardware virtualization is a technology that organizations are widely adopting to enable better utilization for available computing resources. Virtualization is accomplished by

installing monitor software known as hypervisor that allows multiple operating systems to be installed concurrently on the same machine with total isolation from each other [11]. Additionally, many hardware capabilities were added to processors to allow better support for full virtualization. With virtualization, physical hardware resources are assigned to each running instance as required, enabling different clients to access them in a cost-effective manner. Virtualization is usually used as a substitute for multi-tenancy due to its ease of implementation and lower costs.

*Figure 2: Convergence of Cloud Computing Enabling Technologies*



### Moving to the Cloud

Migrating to cloud computing is not a trivial task. The cloud is a different model that both techies and non-techies are not used to working with. Therefore, organizations should be well-prepared for this shift. As illustrated in Figure 3, successful migration process should contain the following steps:

- **Education:** Early adopters should first learn about the basics of cloud computing. Many workshops, conferences, magazines, forums, and case studies are now available to give beginners (both IT and non-IT practitioners) materials and information needed to understand this new paradigm.

- **Needs Assessment:** Cloud computing is by no means a silver bullet, but might be a way to help businesses overcome the limitations of on-premises solutions. Projects should not be driven by the hype; rather, organizations should know exactly why they are moving to cloud computing and what is expected from the switch. It is important that implementers know which parts of their datacenters should migrate to the cloud. It is equally important that they know if this migration is a strategic or tactical decision.

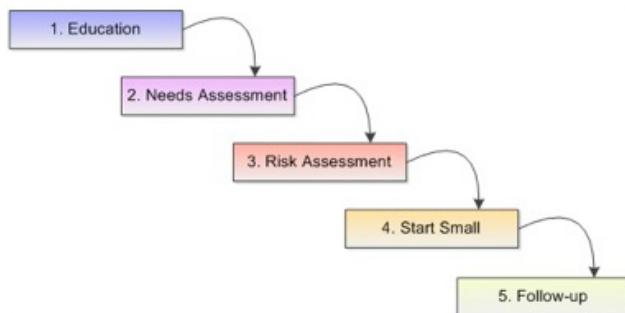
- **Risks Assessment:** As described in the “Cons” section, cloud computing is not a risk-free technology. Adopters should analyze the pros and cons of utilizing cloud computing versus on-premises model in reference to their needs to make sure that risks do not outweigh the benefits.

- **Start Small:** Implementers should not ship all IT projects to the cloud at once—it is not an all-or-nothing decision. Rather, they should start with one small project like those used by small

offices/departments. Implementers will need to learn how to use resources and services of selected provider(s). Developers should learn provider's APIs to allow their applications to dynamically scale up or down their usage in accordance with actual needs. Administrators should know how to manage and monitor used services. In practice, first implantation will come out with a list of lessons learned that can be usefully applied to future projects. This strategy will help organizations get hands-on experience as well as minimize risks associated with the decision to adopt a new technology.

- **Follow-up:** The purpose of this phase is to improve the overall quality of implemented projects. Organizations should assess their projects to decide whether to keep on using cloud option or not. If adopters decide to retain cloud computing, they should continuously review their implementations and decide which parts should stay on the cloud and which should not. During this phase, new cloud projects may be implemented, more data may move to the cloud, and some projects/data may move back from the cloud.

Figure 3: Cloud Computing Migration Steps



## Case Studies

A number of case studies have been published both on providers' websites and in technical reports to give new adopters an inside look at some scenarios that led enterprises to adopt cloud computing and the benefits gained from that turn. A small list of cloud computing examples in different sectors is presented below [12]:

### In SMEs:

- Razorfish, a digital advertising agency, needed to improve its ability to quickly respond to customers demands to support both highly visible web campaigns and high volume short run campaigns. Razorfish employed Rackspace infrastructure solutions to be able to build micro sites, web pages, and blogs more cost effectively. Cloud computing allowed Razorfish to set up web hosting space in 24 to 48 hours rather than 6 to 8 weeks for about \$3,000 to \$5,000 rather than tens of thousands of dollars.

### In Large Enterprises:

- JohnsonDiversey, a global provider of commercial cleaning solutions for business, was motivated to move to the cloud for two reasons. First, to allow better collaboration and integration

between its systems that was hard to accomplish with its legacy on-premise systems. Second, inefficiencies resulted from storage limitations. JohnsonDiversey adopted a number of cloud solutions such as Gmail to replace in-house e-mail; Google Docs to replace Microsoft Office environment; Google Sites for team collaboration; and Oracle CRM On Demand for remote sales force. Cloud solutions allowed JohnsonDiversey to cut operating costs of e-mail and collaboration environment by 70%; reduce bandwidth consumption for messaging and collaboration by 20%; and increase user satisfaction by more than 25%.

### In Government:

- Japan's Ministry of Economy, Trade and Industry needed to build a public web application to enable clients to exchange old appliances for credits toward new appliances and merchandise. This application was planned to work fine for high-scalability requirements to support potentially large transaction volumes—40 million consumers were expected to access the site at peak times. The ministry was able to build the needed application in only three weeks by utilizing Salesforce.com sites and a Force.com API.

## Pros

Cloud computing as a business and technical model derives many of its benefits from other terminologies such as economies of scale, distributed computing, and SOA. These benefits are on hand to both providers and clients.

### Provider Benefits:

- **Better Hardware Utilization:** In most organizations, hardware resources rarely operate at full capacity; consequently, the value of these resources is extremely minimized versus the cost paid to obtain them. Cloud computing can help organizations with large investments in hardware resources to lease unused parts to others.

- **Higher Revenues:** It gives specialties that never existed before in the market the chance to run new businesses that make high incomes. Furthermore, the ability to lease unused hardware resources gives organizations the ability to make extra profits that could be exploited to run and enhance their IT infrastructure.

- **Bigger Software Markets:** Software vendors can deliver their applications in a form of services to their clients at lower costs on a subscription basis. This feature could encourage clients to increase their use of these applications, which in turn, would minimize the rate of software piracy, allowing providers to gain higher revenues.

- **Activities Monitoring:** Providers are able to monitor actions and activities performed by their clients. In doing so, providers can promote other services and products to clients with opportunities to make more money.

- **Better Release Management:** SaaS providers are freed from sending different patches, releases, and upgrades to each single client separately. Given that all software applications are being hosted on provider servers, updates can be instantly and automatically applied without client intervention.

### Consumer Benefits:

- **Reduced Costs:** Cloud computing enables SMEs to have low cost startups by allowing them to rent resources offered by cloud providers instead of having their own sets. Also, large enterprises can take advantage of cloud computing as a tactical solution to face seasonal peaks without spending big sums to acquire resources that will be idle for most of the time. Operational expenses including salaries and energy costs are equally reduced for both small-to-medium and medium-to-large corporations.

- **Reduced Setup Time:** Organizations can acquire and operate necessary resources in almost no time versus much time needed to plan, buy and install their own resources.

- **No Installation/Upgrade Hassles:** With on-premises, organizations spend much time and effort to setup and run IT resources. Conversely, cloud computing put all these complexities on provider sides enabling clients to easily operate hardware and software appliances. Additionally, fixes and upgrades are all made by providers giving their clients the chance to focus on the business.

- **Higher Scalability:** Organizations can effortlessly install any number of hardware/software instances wanted by business. Additionally, clients can freely delete unused instances to save costs. This elasticity gives adopters two main advantages over on-premises models. First, it frees organizations from spending high up-front costs on IT resources that may not be fully utilized in the future. Second, it allows them to face occasional spikes by flexibly adding more resources at whatever time needed.

### Cons

Cloud computing is still in its early years. Organizations usually prefer to adopt proven methodologies that come with success stories and best practices from previous adopters. Some of the risks of adopting cloud computing include:

- **Standards:** Cloud computing lacks the standards needed for loose coupling between providers and clients. Each client should use APIs offered by providers in order to allow its application to make use of available services. That is to say, each provider has its own technologies and standards making it impossible for clients to move from one provider to another.

- **Dependability:** The first question that every client usually asks about adopting cloud computing is, "Is the cloud provider going to be around in future?" Can they get their mission critical information, and is there a way to use it somewhere else? Organizations do not want to invest in IT solutions that may depart with important information if cloud providers decide to leave the market.

- **Transparency:** Because providers have full control over cloud resources, they can make changes to the infrastructure and services without notifying their clients. These issues must be stated in SLAs to guarantee continuity and reliability of solutions used by the clients.

- **Security:** Organizations cannot imagine hosting mission critical information beyond their borders. They believe that losing physical access to and control of servers that host such information means losing information itself. Such an issue makes sensitive information vulnerable to security breaches and surveillance activities of intelligence agencies and/or business competitors.

- **Internet Connections:** Since cloud computing relies on the Internet to host information, having reliable, redundant,

and high-speed Internet connections is critical to successful implementations. Although broadband is available to many parts of the world, some countries still do not have dependable access to the Internet. Another concern related to this point is that although small/micro organizations can have Internet access, they cannot afford having multiple Internet service providers for service availability and reliability. Saving money resulting from leasing resources rather than buying them can be lost on redundant Internet connections and bandwidth. These limitations undoubtedly make it impossible for some organizations to move to the cloud.

- **Availability:** This is a crucial requirement for business stability and success. Key cloud providers invest several hundred million dollars in their hardware resources to guarantee the high level of service provided to their clients. Redundancy of data-centers owned by providers is an essential strategy followed to assure reliability of offered solutions. However, availability and reliability of cloud services are not 100% guaranteed due to unmanaged circumstances. For instance, an Internet connection may be lost for some reason, server(s) crashes may happen on the provider side, human error may cause servers to go down, etc. Lack of availability encourages organizations to locally backup their information for emergency use during cloud outages. Of course, local backup may not be an affordable solution for smaller organizations as it adds more overall cost.

- **Legislation:** Laws related to cloud computing issues such as reliability of presented solutions, availability of providers, and secrecy of information, as well as providers' financial rights, are still missing. Moving to cloud computing depends a great deal on trust between providers and clients and vice versa. With strong and effective legislation, trust between cloud implementers can be built and sustained.

### Alternative Models

Cloud computing is not the only available model in the market that allows organizations to host and run their systems/data on remote servers. Some competing models are briefly described below:

- **Dedicated Servers:** As its name indicates, different clients can lease servers dedicated for their use by hosting companies for defined a length of time. Clients define the specifications of needed servers or they choose from hardware packages with the ability to customize according to their needs (upgrade, downgrade, install applications, etc.). Usually, companies offer dedicated servers in different options; for example, they may entirely/partially manage these servers by their own staff or not. Technically, the main difference a cloud computing model and a dedicated server model is as follows: with dedicated servers, clients are billed for the period they leased the servers and not for the actual use, whereas, cloud computing adopts a utility model that allows clients to pay only for the resources they used during a fixed period.

- **Virtual Private Servers (VPSs):** This solution leverages the capabilities of hypervisors to provide clients with a less expensive form of dedicated servers. In this model, hosting companies split each physical server into a number of virtual instances to be used by different clients concurrently. Each of these instances can run any application that is supported by the host operating system. Clients can also link different VPSs together

to act as a farm/cluster to obtain higher performance needed by heavy traffic websites/applications. Although this model has many similarities with an IaaS layer, they are not identical. In the VPS model, physical servers are usually sold to many clients with no real isolation between them. Those clients are offered instances composed of limited hardware resources compared to IaaS offerings. Thus, these oversold servers can easily lead to poor performance or even system crashes.

• **Colocation Centers:** This model allows clients to host their servers without the burden of having their own datacenters. Co-location centers (aka colos) are being widely used by SMEs that cannot afford building huge and complicated datacenters. With colos, enterprises can easily locate their servers at datacenters powered with spaces, electricity, cooling systems, fire protection systems, communication links, and security strategies. In addi-

tion, some colos offer technical expertise needed to manage servers of clients who cannot hire required human resources.

### Conclusion

This paper presented essential terms related to cloud computing with the aim to answer questions frequently asked by people who are in the computer field. These terms included its history, definition and characteristics, architecture, enabling technologies, key adoption steps, a number of success stories, benefits to both providers and clients, challenges to adopt, and finally a list of top alternative models.

### Acknowledgment

The author thanks Cybèle Cochran for reviewing this paper, and helpful discussions and comments. ♦

## REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing". University of California, Berkeley, Feb 2009
2. H. R. Motahari-Nezhad, B. Stephenson, S. Singhal, "Outsourcing Business to Cloud Computing Services: Opportunities and Challenges". HP laboratories, 2009.
3. J. McCarthy, "Reminiscences on the History of Time Sharing", Stanford University, 1983, <<http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html>>.
4. M. A. Smith, R. L. Kumar, "A theory of application service provider use from a client perspective". Journal of Information and Management, 2004.
5. L.F.G. Sarmenta, "Volunteer Computing". Ph.D. thesis, MIT, March 2001, <<http://www.cag.lcs.mit.edu/bayaniran/>>.
6. L, Wag, G. V. Laszewski, A. Younge , X, He , M. Kunze , J. Tao , C. Fu, "Cloud Computing: a Perspective Study". Journal of New Generation Computing, Volume 28, Number 2, April 2010.
7. I. Foster and C. Kesselman, "The Grid: blueprint for a new computing infrastructure". Morgan Kaufmann, 1998.
8. P. Seltsikas, W. L. Currie, "Evaluating The Application Service Provider Business Model: The Challenge of Integration". Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.
9. Q. F. Hassan, "Aspects of SOA: An Entry Point for Starters". Annals Computer Science Series Volume 7, Issue 2, November 2009, <<http://anale-informatica.tibiscus.ro/download/lucrari/7-2-12-Hassan.pdf>>.
10. S. Murugesan, "Understanding Web 2.0," IT Professional, Volume 9, Number 4, July/Aug. 2007.
11. "Understanding Full Virtualization, Paravirtualization, and Hardware Assist". VMware, <[http://www.vmware.com/files/pdf/VMware\\_paravirtualization.pdf](http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf)>.
12. Case Studies in Cloud Computing, Gartner, <[http://www.gartner.com/it/content/1286700/1286717/march\\_4\\_case\\_studies\\_in\\_cloud\\_computing\\_dcearley\\_gphifer.pdf](http://www.gartner.com/it/content/1286700/1286717/march_4_case_studies_in_cloud_computing_dcearley_gphifer.pdf)>.

## ABOUT THE AUTHOR



**Qusay F. Hassan** is a Ph.D. student in Information Sciences at Faculty of Computers and Information Systems, Mansoura University, Egypt, where he received a BS and a MS in Information Sciences. His research interests include Software Engineering, Web Services, SOA, Distributed Systems, Grid Computing and Cloud Computing. He has authored and co-authored a number of papers and articles that have been published in international journals and magazines. Mr. Hassan also works as a senior software engineer for the United States Agency for International Development (USAID) in Cairo.

**Qusay F. Hassan**  
**Faculty of Computers and Information,**  
**Mansoura University, Egypt**  
**[qusayfadhel@yahoo.com](mailto:qusayfadhel@yahoo.com)**

# A Comparison of Parametric Software Estimation Models Using Real Project Data

George Stark, IBM Global Services

## Section 1: Introduction

Defense project managers and software engineers are often called upon to produce effort, duration, and quality estimates for a new project based on the project's initial needs statement. Often the manager or engineer is solely responsible and accountable for producing and delivering these estimates; in other cases, a senior executive may ask them to develop estimates for his or her use.

Depending on the time available (usually short turnaround—one or two days—is required), the level of uncertainty associated with the project scope (often only a general vision or statement of capability), and the phase of the project (early concept is most common), estimators often rely on one or more rules-of-thumb to arrive at their estimate. Most published models have guidelines for these rules, but there is little empirical data to show how well they work. This paper provides that empirical data for one organization's software development approach.

Project estimation involves translating a set of business objectives or requirements into a measure of product "size." This size measure is then used to estimate the effort, duration, and quality of the final software product. The ability of a system engineer or project manager to align the business objectives with the technical estimates leads to well informed business decisions. The time to complete an estimate is often sufficient only for the use of "rules-of-thumb" for simple models to generate "ballpark" estimates that can then be refined as the project unfolds. Several authors, including Boehm [1,2], Jones [3,4], Rone [5], and others [6–10], have published approaches to arrive at software development effort, project duration in calendar time,

and quality as measured by defects discovered prior to release. In all cases the authors advocate for calibrating their approaches based on data from the organization's previous projects ... but what if this data is not available? The questions, then, are: (1) How good are these approaches "out of the box" using the parameters from the model author's environments?; and (2) Can we rely on them to make business decisions?

### Several authors have contributed critiques and comparison papers of the various models:

- Atkinson, et al. [11] and Pearse, et al. [12] show how simple software metrics, both actual and estimated, can be used to effectively manage the final stages of software development, but they do not address early project estimators.
- Kemerer [13] concluded that metrics-based software project estimation is a viable approach as long as the models are calibrated for the environment. He also concluded that function point-based size estimates were better than source lines of code for the 15 projects studied from one environment.
- Jorgensen and Sheppard [14] found that more than 50% of estimation articles try to build, improve, or compare regression-based estimation models. In further studying expert judgment estimation, they also identified a lack of in-depth studies on the actual use of the approach and real-life evaluations published as journal papers.
- Fenton and Pfleeger [15] concluded that single models may work well in environments for which they were derived, but do not translate well to other environments because of the availability of parameter drivers early on in the estimating process. They recommend changes to estimation model structure and standardization of local data definitions to reduce input subjectivity.
- Jorgensen [16] also reported that expert judgment is the predominant estimation technique used in industry today. He analyzed 15 studies comparing model-based and expert-based effort estimation. The results were a tie: Five in favor of model, five in favor of the expert, and five had no difference. Thus, there was no clearly superior approach to effort estimation.

These results encouraged us to continue to search for a parametric approach that would help us to quickly create a reasonable bound on the effort, duration, and quality for a particular project request. Our investigation identified several candidate models that a systems engineer could apply with little to no historical data. But, just because we could ... should we?

The findings presented herein pertain to our particular collection of 54 completed projects by a well-established and measured software development organization. It is possible that some of our findings are not generally applicable, so practitioners are encouraged to run their own tests and determine which of the available models is best for their particular environment. Nevertheless, most of the results presented here are consistent with our intuition and the conclusions above. Thus, we believe the results provide a good perspective of the value of metrics-based software project estimators and the corresponding rules-of-thumb provided by their inventors.

The next section provides an overview of the various estimating models we tested (four Effort prediction models, three Dura-

tion prediction models, and two defect prediction models). This is followed in Section 3 by a description of the projects included in the analysis and the data collected. Section 4 presents the comparisons and the findings.

## Section 2: Estimating Models

The following subsections describe the models that we evaluated.

### Effort Models

Three common formulas for estimating the effort (in person-months) are based on delivered source lines of code (SLOC). Rone developed a model at IBM based on observations from a variety of software projects for customers, including the space shuttle flight software, school district management, point-of-sale systems, help desk systems, and others. The model has the form:

$$E = (((SLOC/\text{productivity})^1.1)^{1.2})^{1.3}$$

Where E is the effort measured in person-months, SLOC is the delivered project scope measured in thousands of lines of code, and productivity is measured as lines of code/development person-months spent between design and functional test. The multipliers are for independent test (1.1), systems engineering and architecture (1.2), and project & configuration management and additional overhead (1.3).

Similarly, Bailey and Basili [6] developed a formula based on 18 large FORTRAN projects. It is expressed as:

$$E = 3.4 + 0.72 * KSLOC^{1.17} \text{ plus or minus}$$

1 standard deviation

Where effort is measured in person-months, and KSLOC is the delivered project scope measured by thousands of lines of delivered code. While the goal of the article was to demonstrate an approach to calibration using step-wise regression, our question is, "Can this model, with its published parameters help deliver a ready-made estimation formula?"

Finally, Barry Boehm's original COCOMO model [1] has an effort formula for three different types of systems: organic systems, semidetached systems, and embedded systems. The formula is:

$$E = 2.4 * KSLOC^{1.05} \text{ (organic systems)}$$

$$E = 3.0 * KSLOC^{1.12} \text{ (semidetached systems)}$$

$$E = 3.6 * KSLOC^{1.20} \text{ (embedded systems)}$$

Where KSLOC represents the delivered project scope measured by thousand delivered source instructions. Boehm's model has 14 factors that allow the estimator to tailor the estimate by +/- 65% using subjective assessments of each factor. This range was used to establish the upper and lower bounds for the analysis performed.

Caper's Jones [3, 4] and the International Software Benchmarking Standards Group (ISBSG) [7, 8] have also published rules-of-thumb formula to help estimators. These formula are based on project size calculated using Jones' function points metric [17]. The effort estimator is:

$$\text{Effort} = \text{Project Size in Function Points} * \text{Productivity}$$

Caper's Jones has derived the staff productivity (measured in function points/staff-hour) as a function of project size, which can be found in [18]. The ISBSG has also empirically derived the productivity rates of project teams developing on various platforms from their project database. These are shown in Table 1.

Table 1: ISBSG Productivity Rates

Platform	10 <sup>th</sup>	Median	90 <sup>th</sup>	Mean
Mainframe	3.2	11.9	34.4	16.8
Midrange	3.8	10.3	30.6	14.1
PC	2.2	7.1	23.1	10.2
Multi	2.6	6.9	22.2	10.7

### Duration Models

Three of the previously cited sources also offered simple estimates of project duration. These are summarized in Table 2, where D is the project duration in calendar months, Effort is Staff-months, FP is Function Points, and C is a constant representing upper and lower bounds for the estimate's rule-of-thumb.

Table 2: Duration Rules of Thump

Source	Rule equation	Parameters
Boehm	$D = 2.5 * (\text{Effort})^C$	$C = [0.32, 0.38]$
Jones	$D = FP^C$	$C = [0.36, 0.46]$
ISBSG	$D = 0.971 * FP^C$	$C = [0.35, 0.50]$

### Quality Models

Both Rone and Jones also offered simple estimates of the quality of the product in terms of the number of defects that should be removed prior to release. The two models are summarized in Table 3, where Q is the expected number of defects, KSLOC is thousands of delivered source lines of code, FP is Function Points, and C is again an empirical constant representing the model's upper and lower bounds.

Table 3: Quality Rules-of-Thump

Source	Rule equation	Parameters
Rone	$Q = KSLOC^C$	$C = [5.4, 11.2]$
Jones	$Q = C * FP^{1.25}$	$C = [0.05, 1]$

## Section 3: Project Data

Fifty-four projects were selected from an organizational repository containing more than 150 projects. In most respects, these projects are typical of an organization assessed to be at Software Engineering Institute (SEI) CMMI® Level 2/3. This means that good practices are followed and that project plans are tailored from the organizational standard process to ensure a good fit between the project goals and activities. Data is collected on all projects and used to track and control the project against the committed plan of record.

The authors had some prior experience with the selected projects and chose the projects from the repository strictly on the basis of the availability of the project data, rather than the data values. The following criteria were used to select the 54 projects analyzed in this study:

- **Completed.** No in-progress projects were included.
- **Recent.** The repository contains projects since 2000. We decided to only use projects completed since the organization demonstrated an SEI CMMI assessment of Level 2 with some organizational characteristics at Level 3. The organization was assessed at Level 3 with some characteristics at Level 4 in 2006.
- **Software-related.** The repository contains hardware and deployment only projects. Since the estimation models are specific to software, we ignored all non-software projects.
- **Life-cycle phase.** For each completed project, the repository includes data from initial project concept through product release. We excluded the concept phase. We included data that went from technical specification through integration test and release.
- **Necessary metrics.** All data items related to size, effort, duration, and quality were available. Some projects in the repository did not record all data items needed to calculate the effort, duration, and quality models, and, thus, could not be used. For example, many projects collected use cases, objects, or story points for the size measure and could not be used with these models.

Thus, after eliminating the unfinished, pre-Level 2, non-software, and project missing necessary data items, we were left with 54 measurable software projects representing a cross-section of the organization's business. That is, there are innovative development projects, commercial product integration projects, and maintenance/enhancement projects of varying sizes, durations, and levels of process tailoring. These projects are multi-platform (e.g., AIX, Linux, Windows), multi-language (e.g., C, C++, Java, KSL, PERL and other scripting languages), and multi-disciplinary (e.g., IT monitoring solutions, help desk ticketing systems, telephony systems, banking systems).

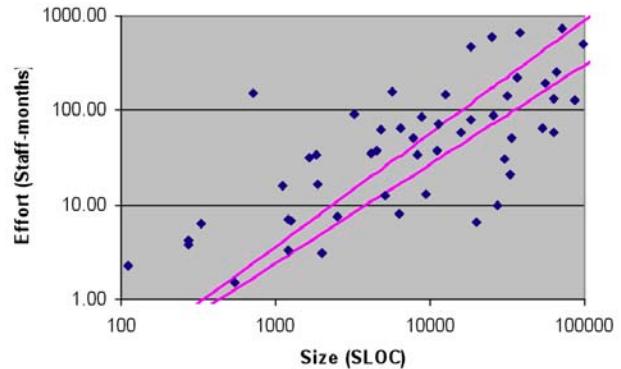
## Section 4: Results

The following collection of plotted graphs show how each of the project estimation models fit the actual data derived from the projects pulled from the historical repository. Each plotted point represents one of the actual projects. Projects plotted above the upper boundary line required more effort than the model predicted; those below the lower boundary line required less effort than the model predicted. Plots with a grey background SLOC as the project content estimator, while those with a white background use the Function Point approach to measure the project content.

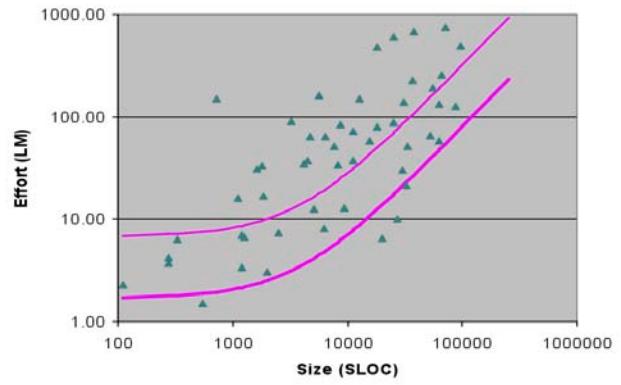
Figure 1 shows the results for the SLOC effort estimation models. The boundaries (for COCOMO I) include the minimum and maximum multipliers for the subjective adjustments available in the model. Using these bounds, only 24% of the projects in our database fell within the bounds of the COCOMO model and more than 46% fell above the upper bound, meaning that the model was very optimistic for our environment. The Bailey and Basili effort estimation model did not fare much better, with 54% of the projects coming in above the upper bound estimate and 33% falling within the bounds. The boundaries on the Rone effort estimation model contained roughly 40% of our projects with 20% of the projects above the upper boundary and the remaining 40% below the lower boundary.

Figure 1: SLOC Effort Estimation models

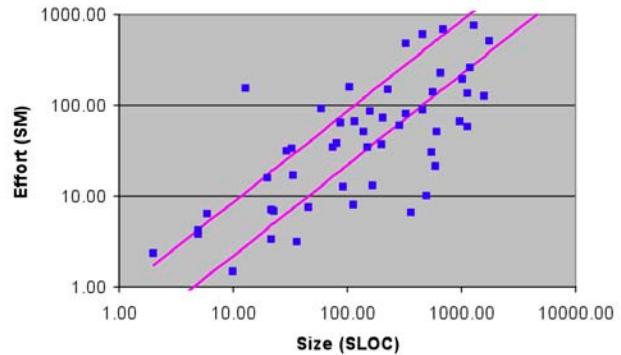
Boehm COCOMO Effort Estimation Accuracy



Basili & Bailey Effort Estimation Accuracy

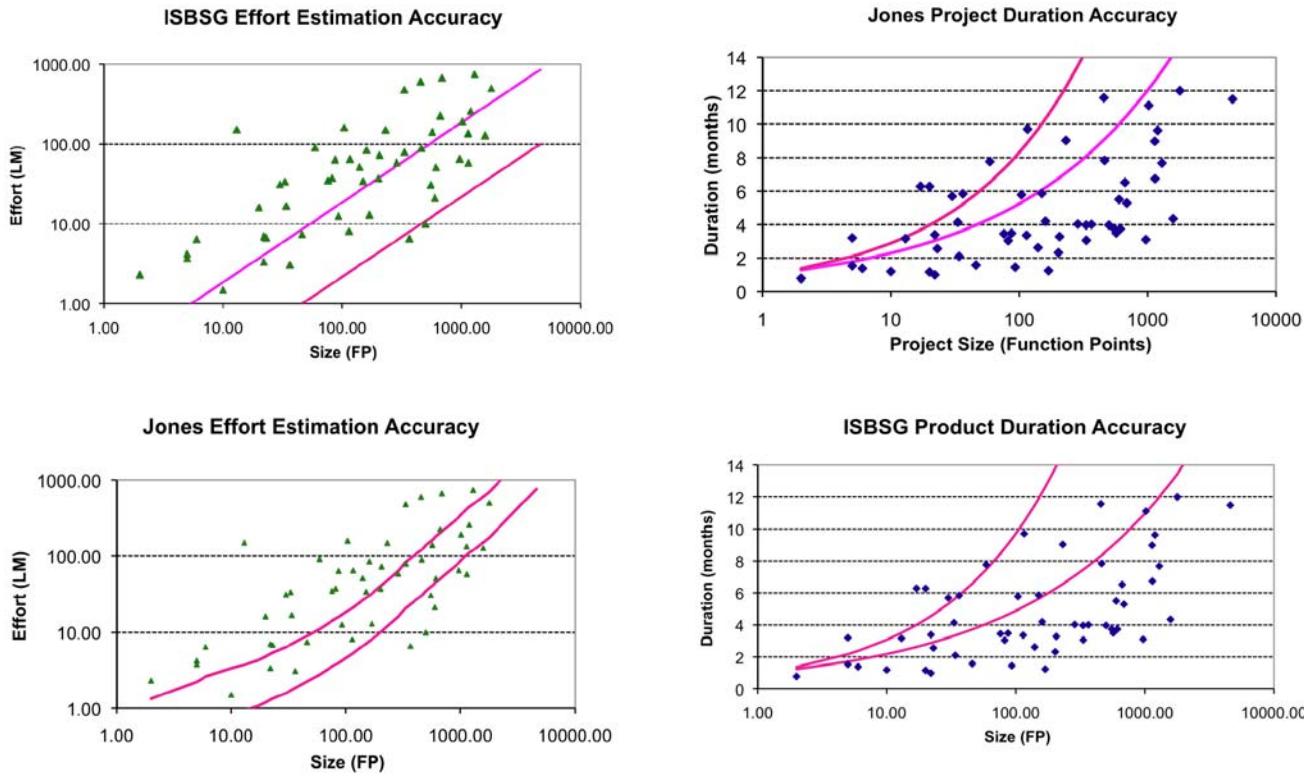


Effort Estimation Accuracy (Rone)



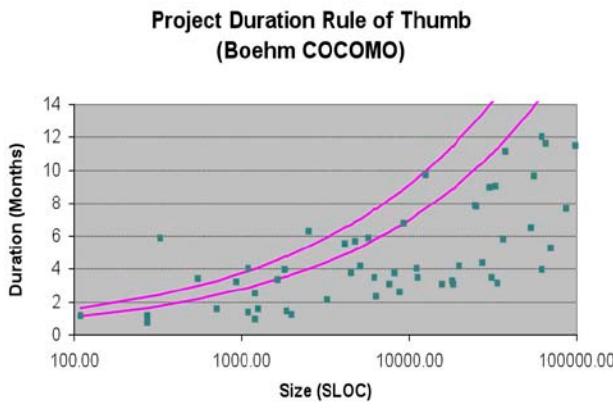
The Function Point effort estimation model results are shown in Figure 2. They did not fare any better than the SLOC models. 50% of our projects were above the Jones upper bound and 63% were above the ISBSG upper bound. In fact, only 13% of the projects were below the lower bound on the Jones model and only 4% of the projects were below the lower bound for the ISBSG approach. This would lead us to conclude that the lower bounds on these rules-of-thumb for effort estimation should never be used in our environment.

Figure 2: Function Point Effort Estimation models



The estimates for duration were significantly better than those for effort. Figure 3 contains the results for those models. Only four of the 54 projects (7%) were above the upper bound associated with the COCOMO-I model, indicating that they took more calendar time than the model estimated. A full 93% were contained below the upper bound and 78% were below the lower bound indicating that it is a cautious estimate to follow early in the process. The ISBSG approach was also quite good with 89% of the projects taking less time than the upper bound using their model and 70% being below the lower bound. The Jones model was comparable with 80% below the upper bound and 75% below the lower bound.

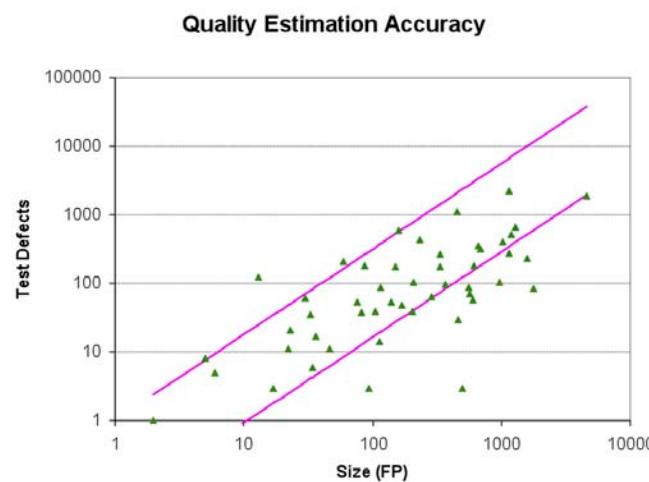
Figure 3: Project Duration Model results



In general, any of these three duration approaches would give our organization a 75% chance of delivering the project on time using the shortest duration estimate.

Figure 4 depicts the results of the quality (i.e., defect) estimates. The Jones model fared well with 60% of the projects falling between the model bounds and 93% of the projects having fewer defects than the model upper bound estimated. The Rone model bounds contained 24% of the projects, with an additional 31% falling above the upper bound estimate.

Figure 4: Quality Estimation Model results



## Summary

Defense managers and system engineers require estimates of project cost/effort, duration, and quality in order to secure funding and set expectations with customers, end users, and management teams. Researchers and practitioners of software metrics have developed models to help project managers and system engineers produce estimates of project effort, duration, and quality. These models generally quantify the project scope using estimated source lines of code or function points, and then require the application of generalized rules-of-thumb to arrive at the needed project estimates of staffing, duration, and quality. Experts agree that for these models to perform at their best, the parameters should be calibrated based on project data from the using organization. Our question was, "How do parametric models perform out-of-the-box (that is, without calibration)?" This is analogous to a project team without access to historical data using the models as published. What level of accuracy can they expect? We examined several published models by comparing the predicted values against the actual results from 54 software projects completed by a SEI CMMI Level 3 organization with a mature (and commended) measurement program.

This paper evaluated a subset of these approaches – nine simple models (four effort estimation models, three duration estimation models, and two software quality (i.e., defect) models) – using 54 non-trivial commercial projects completed recently by a CMMI Level 3 organization. This certification means that the data was collected in a standard manner and makes sense to use in this study. It does not imply that a defined process level is needed to use the results.

For the effort estimation models, we found that the upper bound of the best case model contained 81% of our projects, that is, four out of five of our projects would use less effort than predicted by the best case model, whereas the average effort estimate across all models contained only 54% of our projects, or a little better than a coin flip if we estimate using the average.

Duration estimates performed significantly better. In the best case model, the upper bound estimate contained 93% of our projects with the overall model average at 91% and the lower bound estimate exceeded the actual duration more than 70% of the time. This means we can out-perform the project duration seven out of 10 times using the shortest duration estimated using the models out-of-the box.

For quality modeling, one of the defect prediction approaches worked quite well, with the upper bound containing 94% of the projects (or 9.4 times out of 10 we will deliver fewer defects than forecast by the model). This information is useful to executives and managers performing early project estimates without detailed analysis of the requirements or architecture as the bounds allow them to quickly respond to customer requests with some level of confidence.

So, if you are asked for a project estimate and do not have access to historical data or well-calibrated local estimation models, there is hope. Based on your available sizing information, you can use these models out-of-the-box with some success as long as you keep these things in mind:

- Caper's Jones approach was the only one that (relatively) accurately addressed all three project management estimation needs for effort, duration, and quality.
- None of the four effort estimation models were particularly effective with our project data, but using the upper bound of the Rone model gives the project team an 80% chance of meeting the effort estimate.
- A project should never commit to the lower bound effort estimates from any of the models we evaluated.
- The duration estimation models are particularly effective with our project data. Using the upper bound of the Boehm model gives a project team a better than 90% chance of completing the project within the estimated calendar time.
- Capers Jones' quality model was the most accurate predictor of quantity of defects in our software development projects.

From our analysis, it appears as though duration and quality models are quite useful, but effort estimation is still problematic. We suggest researchers investigate other approaches to effort estimation that are not based on SLOC or Function Points. For example, models that rely on use cases or story points and can estimate all three key parameters (i.e., effort, duration, and quality) may prove valuable in the future. The translation from mission or business need to requirements and architecture is a huge challenge that impacts estimates on each iteration, by developing models to address these early solution descriptions, managers and system engineers can benefit with earlier estimates.♦

### *Disclaimer:*

*The opinions and conclusions are those of the author(s) and not of IBM.*

<sup>®</sup>*CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.*

## ABOUT THE AUTHOR



**George Stark** is an IBM Senior Technical Staff member with over 25 years of experience in software and service measurement and statistical modeling. He has published more than 40 technical papers in referred journals and conferences and has been on the editorial board of the Software Quality Journal. He is currently a member of the Delivery Excellence team where he consults with IBM quality and productivity improvement teams worldwide and is a key leader in the IBM Estimation Community of Practice. He also works with clients on project estimation, software reliability and process improvement approaches.

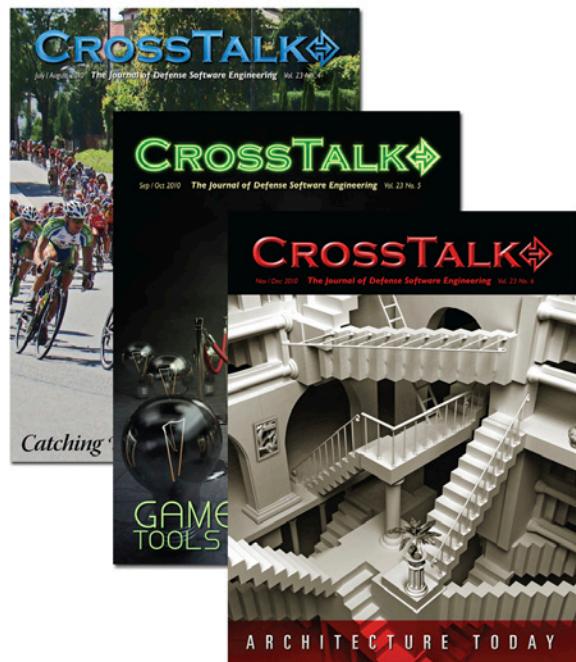
**George Stark**  
**10033 Circleview Drive**  
**Austin, Tx 78733**  
**(512) 653-5438 phone**  
**(512) 263-5024 fax**  
**gstark@us.ibm.com**

## REFERENCES

1. Boehm, B. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ, Prentice Hall.
2. Boehm, B. (2006). "Minimizing Future Guesswork in Estimating," IBM Conference on Estimation, Atlanta, Ga. Feb. 2006.
3. Jones, C., (2007), "Software Estimating Rules-of-Thumb," <<http://www.compaid.com/caliinternet/ezine/capers-rules.pdf>>, Mar. 2007.

## REFERENCES - CONTINUED

4. Jones, C., (1997), *Applied Software Measurement*, 2nd Ed, McGraw-Hill, NY.
5. Rone, K., et al, (1994), "The Matrix Method of Software Project Estimation", proceedings of the Dual-Use Space Technology Conference, NASA Johnson Space Center, Houston, TX, Feb.
6. J.W. Bailey and V.R. Basili, "A Meta-Model for Software Development and Resource Expenditures," Proceedings of the 5th International Conference on Software Engineering, New York: Institute of Electrical and Electronic Engineers, 1983.
7. ISBSG, International Software Benchmarking Standards Group, <<http://www.compaid.com/caliinternet/ezine/ISBSGestimation.pdf>>
8. ISBSG, International Software Benchmarking Standards Group, <<http://www.isbsg.org/isbsg.nsf/webs/Project%20Duration>>
9. McConnell, S., (2006), *Software Estimation: Demystifying the Black Art*, Redmond, WA, Microsoft Press.
10. P. Oman, "Automated Software Quality Models in Industry," Proceedings of the Eighth Annual Oregon Workshop on Software Metrics, (May 11-13, Coeur d'Alene, ID), 1997.
11. G. Atkinson, J. Hagermeister, P. Oman & A. Baburaj, "Directing Software Development Projects with Product Measures," Proceedings of the Fifth International Software Metrics Symposium, (Nov. 20-21, Bethesda, MD), IEEE CS Press, Los Alamitos, CA, 1998, pp. 193-204.
12. T. Pearse, T. Freeman, & P. Oman, "Using Metrics to Manage the End-Game of a Software Project," Proceedings of the Sixth International Software Metrics Symposium, (Nov. 4-6, Boca Raton, FL), IEEE CS Press, Los Alamitos, CA, 1999, pp. 207-215.
13. Kemerer, C. F., (1987), "An empirical validation of software cost estimation models," *Communications of the ACM*, Vol 30, No 5, pp. 416-429.
14. Jorgensen, M., and Sheppard, M., (2007), "A Systematic Review of Software Development Cost Estimation Studies," *IEEE Transactions on Software Engineering*, Vol 33, No 1, Jan. pp 33-53.
15. Fenton N. E., and Pfleeger, S. L., (1997), *Software Metrics: A Rigorous & Practical Approach*, 2nd Ed., London, PWS Publishing.
16. Jorgensen, M., (2004), "A Review of Studies on Expert Estimation of Software Development Effort," *Journal of Systems & Software*, Vol 70, no 1, pp. 37-60.
17. International Function Point User's Group (IFPUG), *Function Point Counting Manual*, Release 3.1, 1990.
18. Jones, C., "Achieving Excellence in Software Engineering," presentation to IBM Software Engineering Group, March, 2006.



## CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, **CROSSTALK** can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

### DoD Gaming and Virtual World Applications

*July/August 2011*

Submission Deadline: February 11, 2011

### Protecting Against Predatory Practices

*September/October 2011*

Submission Deadline: April 8, 2011

### Software's Greatest Hits and Misses

*November/December 2011*

Submission Deadline: June 10, 2011

Please follow the Author Guidelines for **CROSSTALK**, available on the Internet at <[www.crosstalkonline.org/submit](http://www.crosstalkonline.org/submit)>. We accept article submissions on software-related topics at any time, along with Letters to the Editor and BackTalk. To see a list of themes for upcoming issues or to learn more about the types of articles we're looking for visit <[www.crosstalkonline.org/theme-calendar](http://www.crosstalkonline.org/theme-calendar)>.

# Green Light Lag, Yellow Light Drag

What lies at the heart of an engineer? What differentiates an engineer from a scientist, architect, craftsman, or artist?

Is it the desire to learn how things work? Perhaps, but an engineer is grounded in the practical; leaving black and worm holes to the lab coats.

Is it the desire to create and design? Perhaps, but few engineers are offered the luxury of the architect's blank design sheet. Our task begins, rather than ends, with design.

The craftsman within an engineer wants to get his hands dirty. Mechanical engineers grind a gear or two. Civil engineers parade their hard hats and steel-toed shoes. Electronic engineers have their soldering guns holstered at the ready. Even software engineers—who produce untouchable products—are often caught looking under the hood of their computers, peripherals and servers much to the chagrin of system administrators. Still engineering is more than tactile tinkering.

Like the artist, engineers see beauty in their creations, however our muse is not aesthetic but functional. We prefer client awe and satisfaction to critical adoration.

At the heart of an engineer is the desire to create solutions. We take pleasure in transforming raw materials into innovative and useful gizmos, thingamajigs, and whatchamacallits.

It starts early in an engineer's life with Building Blocks, Tinker Toys and Lincoln Logs. You quickly graduate to Legos and Erector Sets. You master the designs on the box and in the instruction manual. Then the real fun comes in creating your own designs.

The desire and thrill never leaves. One night I conjured up a new game with my son Matt. I went to the pantry and pulled out a package of plastic cups and divvied them out evenly between the two of us. The challenge? Build the highest cup tower with the least amount of cups. Raw materials, strategy, skill, and trade-offs; heaven.

My latest toy? Buckyballs! Not the spherical fullerenes but a set of 216 magnetic balls. Each five-millimeter ball has a neodymium core charged to a magnetic flux of 50. The small

size and strong magnetism allows one to create interesting patterns that form building blocks for larger objects. Combining objects changes their polarity and adds design dilemmas and opportunities.

Such toys awaken an engineer's inner master builder to solve straightforward problems. Straightforward problems are not necessarily simple or easy but their solutions are characterized by a set of instructions that can be taught and repeated with equal success.

For example, if I give you a single string of 216 Buckyballs and ask you to form a 6x6x6 cube it would take you several hours. It's not intuitive or easy. However, if I spent 5 minutes teaching you step-by-step how to form the cube from the strand, including some techniques in handling Buckyballs you could master the cube and teach others.

Early engineering education focuses on straightforward problems that can be mastered, repeated and shared. However, engineers are not hired to execute recipes; we are hired to solve complex problems.

Complex problems increase in scale, interconnectivity and discipline. They tend to overwhelm the inner master builder's skill and capacity. Complex problems have to be broken down into subsets of straightforward problems and solved by specialists in various disciplines. Building the next generation fighter jet requires a variety of engineers and craftsmen working in concert. To succeed at complex problems engineers need to evolve from master builders to conductors where timing and coordination are critical. This transition is sensed in a phenomenon I call the Green Light Lag.

Picture yourself alone at a traffic light when it turns green. What happens? You start moving immediately with no delays. Now picture yourself ten cars back at the same traffic light. Now what happens when the light turns green? It takes eight to 10 seconds before you can move your car. Why?

Theoretically when the light turns green all 10 cars should start rolling simultaneously without mishap. If the Thunderbirds and Blue Angels can move tightly together in high speed 3-D why

can't we do so in low speed 2-D? What happened? Due to a lack of trust, preparation, practice, communication and coordination each car in line waits for the car ahead to move before acting thus contributing to the green light lag.

In the same way, if a project team lacks trust, preparation and coordination the project will suffer green light lag. It is the role of the conductor, the engineer leading the project, to build team trust, open communication and reinforce coordination. The evolution from a straightforward engineer (builder) to a complex engineer (conductor) requires the development of skills engineers do not bargain for when they entered the field like scheduling, negotiation, collaboration and leadership skills.

With the explosion of the information age we now face yet another level of problems. This new breed of problems are not only complex but animated, volatile, fickle, and ever changing. In a word they are mercurial.

Mercurial problems require a conductor to increase trust by pushing most decisions to the periphery, giving super specialists the room to adapt to rapid changes and unexpected problems based on their expertise. In turn this puts more emphasis on effective coordination.

Mercurial problems are particularly prevalent in software engineering. You build a complex payroll system for a mainframe processor only to be told the system now needs to run on multicore processors with a browser interface taking advantage of a parallel design. Can your engineers handle that? What do you do?

With mercurial problems your old designs, processes, and workforce often fall short. Sure you can protest requirements creep, lack of a stable baseline, or unrealistic expectations, however, in the future successful engineers will be mercurial engineers—adaptive, animated, lively, and quick-witted. They will augment their building and coordination skills with the assemblage of highly specialized, skilled and adaptive teams.

Back to the intersection. There is another phenomenon that occurs when the traffic light transitions from green to yellow. Tepid drivers impulsively stop and wait. Lackadaisical drivers are caught by surprise, brake late and block the intersection. Vigilant drivers do the yellow light drag—accelerating through the light before it turns red.

Where are you driving your engineering skills; green light lag, yellow light drag, or blocking the intersection?

**Gary Petersen**

Arrowpoint Solutions, Inc.

# Web Exclusives

As with every issue, we had a litany of great articles submitted for this issue. **CROSSTALK** would like to publish them all, but unfortunately we are limited by space and layout restrictions. **CROSSTALK** is proud to announce that, although we are limited in print, we are not necessarily limited online. Frequently you will see additional great articles for each issue posted online as web-exclusive features. For the January/February 2011 issue, you can look forward to the following web exclusive:

## Design Point:

### An Empirical Approach for Estimating Design Effort

**Abstract.** In this paper, we present an extension to Function Point estimation, Design Point, conceived to estimate size and productivity of design phase for software development projects executed by Infosys. This approach is based on capturing functional and non-functional requirements, identifying design sensitive parameters influencing the design phase, and deriving design size for any development project. An empirical validation and refinement of model (identification of design sensitive parameters and degree of influence for each of the parameters) has been performed to test the hypothesis over a large number of projects in different stages of execution.

**By Srinivasan Venkataraman, Pratip Sengupta, Amit Patni, and Bibhash Saha**

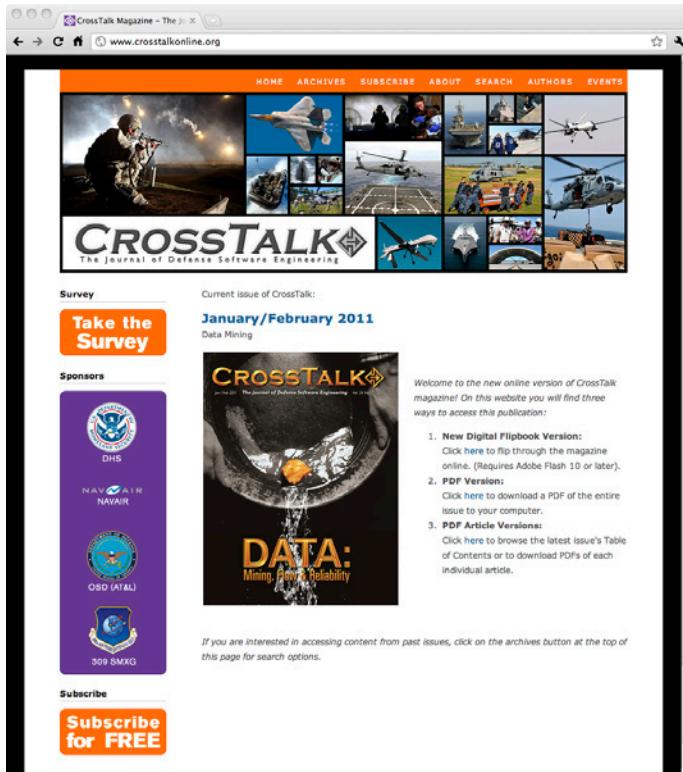
Be sure to check future issues online at <<http://www.crosstalkonline.org>> for more web exclusives!

**Justin T. Hill**

Acting Publisher

**CROSSTALK**, The Journal of Defense Software Engineering

# Reminder: **CROSS**TALK Now Online-only



As a reminder, **CROSSTALK** is now completely electronic. New issues will be posted six times a year on **CROSSTALK**'s new website, <<http://www.crosstalkonline.org>>. Please update your browser's bookmarked **CROSSTALK** URL to reflect the new web address. If you are currently subscribing to **CROSSTALK**'s RSS Feed, please note the feed URL has also changed to <<http://www.crosstalkonline.org/Issues/rss.xml>>.

Each new issue will be available online both as a downloadable PDF file and also as a Flash-based digital flipbook viewable within a browser and designed to mimic the look and feel of a printed magazine

This change reduces our carbon footprint and allows us to bring the Journal of Defense Software to our readers in their preferred and most convenient formats. This is also **CROSSTALK**'s first step towards reaching new reader devices and enhancing the suitability of the journal for our increasing electronic readership.

To help guide the transition to other digital formats, we have posted a brief reader survey. Please take a moment to participate in the survey by clicking on the "Take the Survey" button on the <<http://www.crosstalkonline.org>> home page or by visiting <<http://www.crosstalkonline.org/survey>> directly. Data gathered from this survey will be used to help determine future **CROSSTALK** digital and mobile formats. Your input into the future direction of **CROSSTALK** is greatly appreciated.

Thank you for your continued support and from all of us at **CROSSTALK**, best wishes for the New Year!

**Justin T. Hill**

Acting Publisher

**CROSSTALK**, The Journal of Defense Software Engineering

23rd Annual

# SSTC

Systems & Software  
Technology Conference

## Topics Include...

### \* Concepts and Trends

#### Example Areas:

- Cloud Computing
- Model Driven Processes
- Multi Processor Challenges
- SOA

### \* Cyber Technologies

#### Example Areas:

- Cyber Security
- Cyber Defense
- Cyber Physical Systems

### \* Guidance, Policies, and Standards

### \* Human Capital / Workforce Development

### \* Modernization of Systems

### \* Real World Lessons

### \* Research

### \* Social Networking

### \* Technological Tool Advances

#### Example Areas:

- Acquisition Processes
- Agile Development
- Assessments
- Data Development & Environments
- Program Management & Methods
- Rugged and Resilient Systems
- Smart Grids
- Systems & Systems Assurance
- Software
- Testing Methods
- Verification / Validation
- Web Authentication

Plan now to join us for excellent, quality presentations and networking with colleagues from military/government, industry and academia.



Conference Registration Opens 24 January 2011

**Mark your calendar!**

For conference & trade show information, visit  
**WWW.SSTC-ONLINE.ORG**

Ok, we've all seen it, individuals totally immersed in hand-held devices consuming an endless stream of information and knowledge - thumbs furiously dancing across minuscule keyboards entering thoughts, ideas, and directions.

Much like the tiny information exchange tools meant to keep us up-to-speed in this data-filled world, SSTC 2011 will focus on connecting attendees with technological advancements associated with building better systems and software in support of our defense forces.

**Join us 16 – 19 May 2011**  
**in Salt Lake City, Utah, as we Sync-Up with advances in Technology!**

# HILL AIR FORCE BASE IS HIRING SOFTWARE ENGINEERS AND COMPUTER SCIENTISTS



## EXCITING AND STABLE WORKLOADS:

- ★ Joint Mission Planning System
- ★ Battle Control System-Fixed
- ★ Satellite Technology
- ★ Expeditionary Fighting Vehicle
- ★ F-16, F-22, F-35
- ★ Ground Theater Air Control System
- ★ Human Engineering Development

## EMPLOYEE BENEFITS:

- ★ Health Care Packages
- ★ 10 Paid Holidays
- ★ Paid Sick Leave
- ★ Exercise Time
- ★ Career Coaching
- ★ Tuition Assistance
- ★ Retirement Savings Plans
- ★ Leadership Training

## LOCATION, LOCATION, LOCATION:

- ★ 25 minutes from Salt Lake City
- ★ Utah Jazz Basketball
- ★ Three Minor League Baseball Teams
- ★ One Hour from 12 Ski Resorts
- ★ Minutes from Hunting, Fishing, Water Skiing, ATV Trails, Hiking

Visit us at [www.309SMXG.hill.af.mil](http://www.309SMXG.hill.af.mil). Send resumes to [shanae.headley@hill.af.mil](mailto:shanae.headley@hill.af.mil).

Also apply for our openings at [USAjobs.gov](http://USAjobs.gov)



NAV AIR



CROSS TALK thanks the above organizations for providing their support.